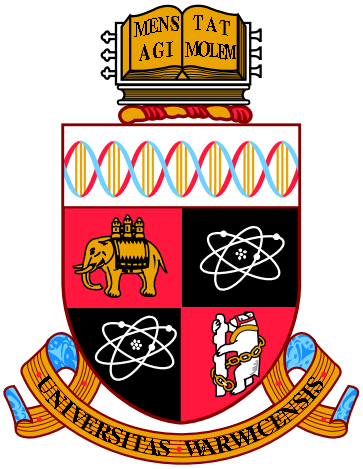


Accelerating Minimum Energy Path Finding Methods

by

Stela Makri

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

School of Engineering

September 2019

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my appreciation and admiration to my supervisors James Kermode and Christoph Ortner. Working under their supervision has been a great pleasure and I am thankful for that. Their continuous guidance and support was invaluable during my studies and their enthusiasm has been a source of inspiration to me. Further, I am grateful to Petr Grigorev for collaborating with us on the results conducted on the screw dislocation system which is discussed in Chapters 2, 4 and 5.

My studies were supported by the Warwick School of Engineering, the EPSRC, the ERC and the Royal Society. During my studies I found great pleasure in exchanging ideas and presenting my work in workshops and conferences. This would not have been possible without the financial support of the Warwick School of Engineering and EPSRC. Computing facilities were provided by the Scientific Computing Research Technology Platform of the University of Warwick with support from the Science Research Investment Fund.

Finally, I would like to thank my family and friends for their ongoing moral support. Especially, my parents who always support my aspirations, my brother and uncle for their troubleshooting support and my late grandfather for his devotion in my educational upbringing and for inspiring me to start this PhD. I am also grateful to my friends Elli and Apostolis for their kind hospitality.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree or diploma at this or any other university. During my PhD studies, I have contributed to the following publication:

- Stela Makri, Christoph Ortner, and James R. Kermode. A preconditioning scheme for minimum energy path finding methods. The Journal of Chemical Physics, 150(9):094109, 2019

on the preconditioning and adaptive time stepping schemes. Material from this publication appear in this thesis. Work conducted on the screw dislocation system included in this thesis were a collaborative work from private communications with Petr Grigorev.

Stela Makri

September 2019

Abstract

The behaviour of physical systems is often characterised by rare transitions between energy minima. Due to their rarity, these events are not observed in typical molecular simulations but can be observed in simulations on the mesoscale provided that the rate of transition is known. For the evaluation of the transition rate, knowledge of the energy barrier the system overcomes during the transition, is necessary.

In Transition State theory this is achieved by finding the saddle point, which lies between the reactant and product states. Popular techniques include the nudged elastic band method and the string method which evaluate the minimum energy path of the transition, as well as the walker-type dimer method. These are iterative schemes evolving a chain of states along projected steepest descent directions. In this thesis, saddle point finding methods are studied with respect to their performance and stability. An adaptive step selection scheme is proposed, motivated from adaptive ODE numerical solving. The scheme improves the robustness of saddle point search methods and enjoys convergence efficiency.

An important factor contributing to slow convergence, comes from not considering potential energy curvature information. This has been addressed for walker-type methods but not for methods which identify transition paths. A preconditioning scheme is proposed to address this issue, which substantially reduces the computational cost of transition path finding algorithms.

Finally, momentum descent methods are explored in conjunction with saddle point finding techniques. A momentum term is introduced in the dynamics of string, nudged elastic band and dimer methods introducing an added inertia which accelerates the dynamics and achieves faster convergence. We demonstrate the improved performance of our approaches in a range of examples including vacancy and dislocation migration modelled with both interatomic potentials and density functional theory.

Abbreviations

ASE	Atomic Simulation Environment
BCC	Body-Centred Cubic
BFGS	Broyden-Fletcher-Goldfarb-Shanno optimisation algorithm
Cu	Copper
DFT	Density Functional Theory
EAM	Embedded Atom Model
EXP	Exponential preconditioner
FCC	Face-Centred Cubic
FF	Force Field preconditioner
FIRE	Fast Inertial Relaxation Engine
GGA	Generalised Gradient Approximations
HCP	Hexagonal Close Packed
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
LBFGS	Limited memory Broyden-Fletcher-Goldfarb-Shanno optimisation algorithm

LCAO	Linear Combinations of Atomic Orbitals
LDA	Local-Density Approximation
MEP	Minimal Energy Path
MM	Molecular Mechanical
NEB	Nudged Elastic Band
ODE	Ordinary Differential Operator
PBE	Perdew-Burke-Ernzerhof functional
QM	Quantum Mechanical
QM/MM	Quantum Mechanical / Molecular Mechanical model
W	Tungsten

Chapter 1

Introduction

In computational chemistry, structural biology, materials science and engineering, the behaviour of processes is often dominated by rare transitions between energy minima of the associated potential energy landscape. Examples include: dislocation creep in crystalline materials, enzyme catalysis, protein folding and fatigue crack growth.

Transitions of this kind are difficult to observe in dynamical simulations. In practice, the system borrows energy from its surroundings, to overcome the energy barrier and escape the attraction of a potential well [4]. To achieve this, many unsuccessful trials take place. However, the typical timescale accessible to simulations is much smaller than the time the system spends in the potential well, before it escapes its attraction. Instead, such transitions can be simulated on the mesoscale, using, for example, the kinetic Monte Carlo method [5], provided that the transition rate is known. An approximation to the transition rate may be found from Arrhenius' law [6],

$$\nu \sim \nu_0 \exp\left({}^{-\Delta E}/_{k_{\mathrm{B}}T}\right), \tag{1.1}$$

in terms of the height of the energy barrier with Eyring's heuristic derivation [7], or Harmonic Transition State Theory [8].

To find the energy barrier of the transition, one needs to find the *transition state*, that is, the saddle point between the two potential wells. There are two families of methods for tackling this problem: methods which require knowledge of the reactant state alone - 'walker-type' methods - such as

the dimer method [9] and methods for which both the reactant and product states are known, the most widely used techniques in which case are the string method [10, 11, 12] and the Nudged Elastic Band (NEB) method [13, 14]. The dimer method evolves a single state of the system, often starting from the local minimum, until it converges to the saddle point, whereas the string and NEB methods search for the Minimum Energy Path (MEP) of the transition - which, loosely speaking, is the most probable path of the transition - by iteratively relaxing a discretised path of images. The MEP passes through the saddle point with probability 1.

This thesis will explore techniques to accelerate the convergence speed of general minimum energy path finding methods, with focus on the NEB and string methods. The dimer method will also be visited. All methods of interest are iterative and evolve a chain of states (or images) along projected steepest descent directions. As a direct consequence, the methods inherit the slow convergence that characterises steepest descent optimisation in the cases where the potential energy is *ill-conditioned*, i.e. the Hessian matrix of the potential has a large condition number [15]. For example this might be the case if there are large discrepancies between bond stiffnesses in the system. A more substantial review of the relevant literature is explored in chapter 2.

The MEP can be computed as the steady state solution of a first order Ordinary Differential Equation (ODE). The same is true for the saddle point in the case of the dimer method. In chapter 3 we see that this allows the use of the theory of adaptive ODE solvers to develop an adaptive step selection scheme for the string, NEB and dimer methods. The adaptive step selection aims to improve not only the performance but also the robustness of the methods. It also benefits from the fact that it can also be used directly for minimisation, as well as saddle point search.

Preconditioning is commonly used in linear algebra and optimisation to effectively reduce the condition number and thus improve the convergence speed of iterative optimisation schemes [15]. It has also been successfully applied to the dimer method [16]. Chapter 4, proposes a novel preconditioning scheme for the string and NEB methods. In essence, the scheme uses potential energy curvature information to treat ill-conditioned geometries and effectively reduce the number of force evaluations required for convergence. Provided that

the cost of constructing and evaluating the preconditioner is much lower than the cost of evaluating the model, the reduced number of force computations leads to substantial speed ups, allowing us to tackle problems that are computationally costly, such as Density Functional Theory models (DFT) containing 10s-100s of atoms. Further, it allows lower residual errors to be obtained than previously possible, improving accuracy and confidence in the solution.

Finally, momentum descent methods are explored in chapter 5. This is a family of two-step methods which introduces an inertia term to the steepest descent method in order to enhance its convergence speed. The computational speedup arises from an improvement in the scaling of the convergence speed with the condition number. We extend this scheme to transition state finding, where preconditioned cases are also explored.

We conclude in chapter 6 where we reflect on our results and discuss the strengths and shortcomings of our proposed work, leading to future research directions.

Chapter 2

Background

In this chapter, we review the preliminary theory required for carrying out this study. We commence by briefly reviewing general concepts of crystal structure simulations and modelling and carry on to define the general problem that we will study as well as the notation used. Additionally, the review covers concepts from numerical analysis which will be useful both for better understanding the problem at hand, but also in our subsequent analysis.

2.1 A note on crystal systems

A crystal, is defined as a solid material whose constituent atoms, molecules or ions are arranged in a periodic array. To describe the geometry of the periodic array, one may refer to the *Bravais lattice*, which is defined as an infinite array of discrete points, such that the arrangement and orientation of points in the array is invariant for all reference points in the lattice [17]. Formally, it is the set of points $\mathcal{R}$ satisfying [18]

$$\mathcal{R} = \Big\{ n_1\boldsymbol{a}_1 + n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3 |\ n_i \in \mathbb{Z}\ \forall i = 1,2,3,\ \boldsymbol{a}_i \in \mathbb{R}^3,\ (\boldsymbol{a}_1 \times \boldsymbol{a}_2)\cdot\boldsymbol{a}_3 \neq 0 \Big\}. \tag{2.1}$$

The vectors $\boldsymbol{a}_i \in \mathbb{R}^3$, which span the Bravais lattice, are called *primitive vectors*.

Of course one should note here that since in the physical world crystals are finite, the 'periodicity' is to be taken in the sense that the crystals are large enough so that most points of the Bravais lattice will be sufficiently far

from the surface as to not feel its presence [17].

In the thesis we will investigate systems from two Bravais lattice geometries, the *Body-Centred Cubic* (BCC) and the *Face-Centred Cubic* (FCC). Before we give a formal definition for each of them, we will first take a look at a simpler case, the *simple cubic* three dimensional Bravais lattice. This is constructed by 3 mutually orthogonal primitive vectors of equal length, say α. For example, it may be generated by primitive vectors

$$\boldsymbol{a}_1 = \alpha(1,0,0), \quad \boldsymbol{a}_2 = \alpha(0,1,0), \quad \boldsymbol{a}_3 = \alpha(0,0,1). \tag{2.2}$$

The simple cubic lattice is shown partially in Fig. 2.1.

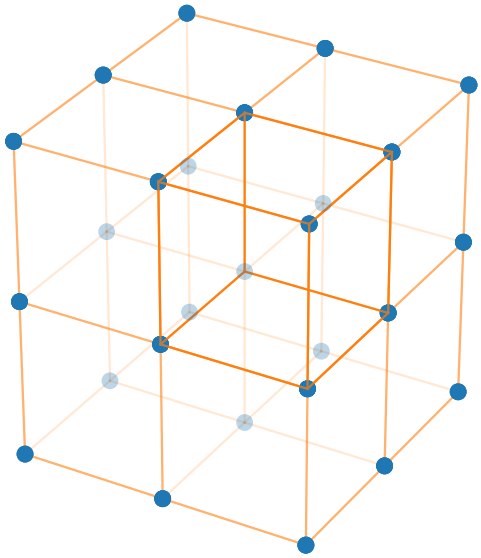


Figure 2.1: Section of the simple cubic Bravais lattice.

Now, for each of the small cubes in the cubic three dimensional lattice, we add another point at the centre. The new lattice formed is the Body-Centred Cubic lattice and it may be constructed with the following choice of primitive vectors

$$\boldsymbol{b}_1 = \frac{1}{2}(\boldsymbol{a}_2 + \boldsymbol{a}_3 - \boldsymbol{a}_1), \quad \boldsymbol{b}_2 = \frac{1}{2}(\boldsymbol{a}_3 + \boldsymbol{a}_1 - \boldsymbol{a}_2), \quad \boldsymbol{b}_3 = \frac{1}{2}(\boldsymbol{a}_1 + \boldsymbol{a}_2 - \boldsymbol{a}_3). \tag{2.3}$$

A simple period of the BCC Bravais lattice is depicted in Fig. 2.2.

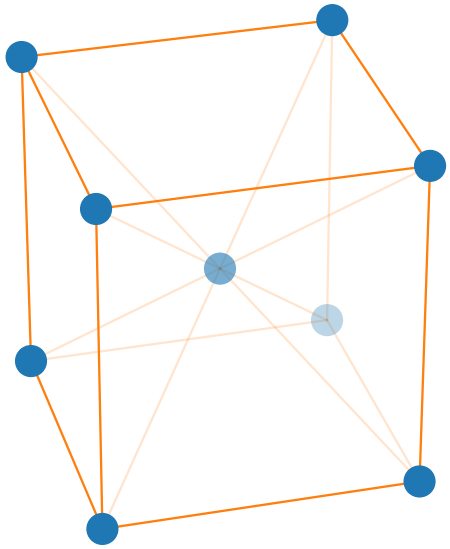


Figure 2.2: Cubic unit cell of the BCC Bravais lattice

To construct the Face-Centred Cubic Bravais lattice, we start from the simple cubic lattice and add an additional point at the centre of each face of the small cubes. A set of primitive vectors for the FCC lattice is made up of

$$\boldsymbol{c}_1 = \frac{1}{2}(\boldsymbol{a}_2 + \boldsymbol{a}_3), \quad \boldsymbol{c}_2 = \frac{1}{2}(\boldsymbol{a}_3 + \boldsymbol{a}_1), \quad \boldsymbol{c}_3 = \frac{1}{2}(\boldsymbol{a}_1 + \boldsymbol{a}_2), \tag{2.4}$$

and the visualisation of the lattice is shown in Fig. 2.3.

In the set of examples we will be studying, we included a copper (Cu) system which has a FCC structure and tungsten (W) systems, having a BCC structure.

2.2 Potential Energy Surfaces

A system's behaviour is governed by a potential energy, V, which is a function of the positions $\{\boldsymbol{r}_n\}$ of its constituent particles. In this section we describe a set of potential energy landscapes which we will meet in our examples.

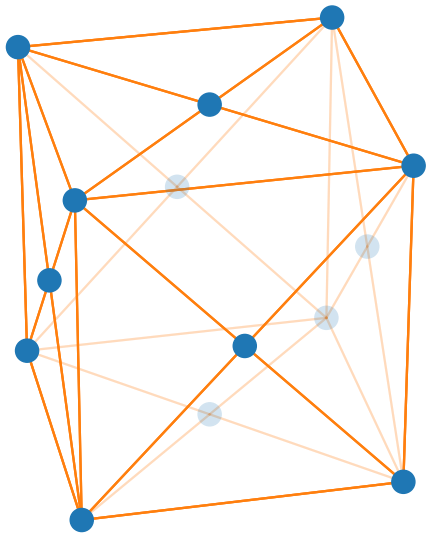


Figure 2.3: Cubic unit cell of the FCC Bravais lattice.

2.2.1 Müller-Brown potential surface

We first look at an artificial energy landscape. The Müller-Brown potential is a simple 2-dimensional toy model, introduced in 1979 by K. Müller and L. O. Brown [19]. This model does not describe any physical model, however, because it is a two dimensional surface, the model allowed Müller and Brown to demonstrate visually their proposition. Since then, it was exploited on many occasion by the theoretical chemistry community to test algorithms often related to reaction-path finding (see Section 2.4). The analytical expression for the potential is:

$$V(x,y) = \sum_{k=1}^{4} \Big\{ A_k \exp\big[a_k(x-x_k^0)^2 + b_k(x-x_k^0)(y-y_k^0) + c_k(y-y_k^0)^2\big] \Big\}, \quad (2.5)$$

where typically the following values are assigned to the constants A, a, b, c, x^0 and y^0:

$$A = (-200, -100, -170, 15), \quad x^0 = (1, 0, -0.5, -1), \quad y^0 = (0, 0.5, 1.5, 1),$$
$$a = (-1, -1, -6.5, 0.7), \quad b = (0, 0, 11, 0.6), \quad c = (-10, -10, -6.5, 0.7).$$

The contours of this potential surface are shown in Fig 2.4. A notable feature of the Müller-Brown potential surface is the fact that it has three local minima, $\boldsymbol{x}_{\mathrm{A}}$, $\boldsymbol{x}_{\mathrm{B}}$ and $\boldsymbol{x}_{\mathrm{C}}$, and two saddle points, $\boldsymbol{x}_{\mathrm{S}_1}$ and $\boldsymbol{x}_{\mathrm{S}_2}$.

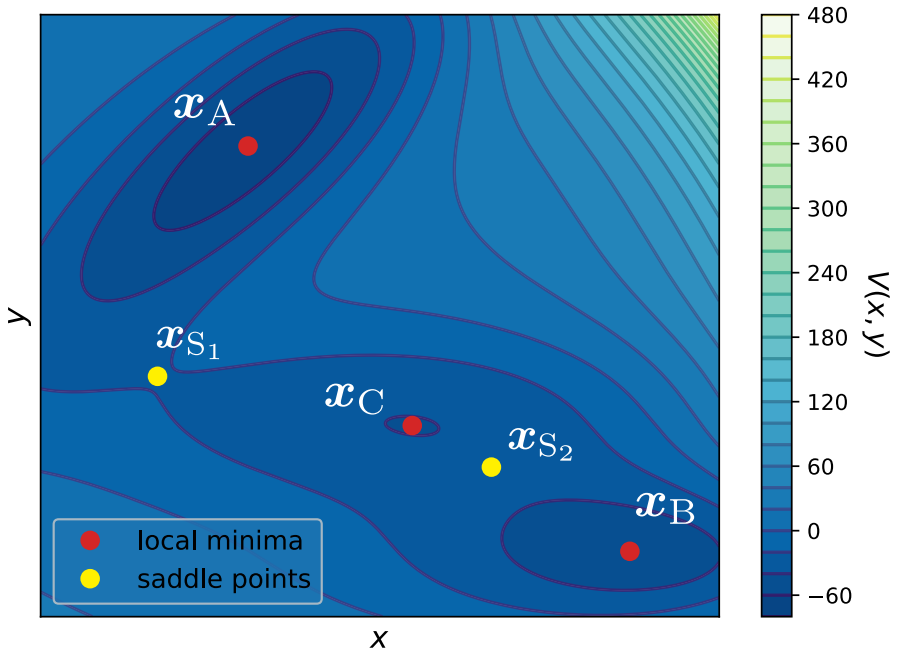


Figure 2.4: The contours of the Müller-Brown potential. The three local minima of the potential are marked in red and the two saddle points in yellow.

2.2.2 Molecular Dynamics

An important tool for studying the behaviour of a system of atoms or molecules, are *Molecular dynamics*. Molecular dynamics were first introduced in 1955 by Fermi *et al.* [20]. These are computer simulations which use Newton's laws of motion

$$-\nabla_i V = m_i \ddot{\boldsymbol{r}}_i \tag{2.6}$$

to describe the trajectories of atoms in a system identified by indices i feeling a total force $-\nabla_i V$ within a given time frame, where V is the potential felt by the system. Energies and forces are often calculated by means of an interatomic potential, but this is not restrictive. All that is required is an expression of the force and associated energy of the system of atoms at any given set of positions

of the atoms $\{\boldsymbol{r}_i\}$ of the system. Solving Newton's equation of motion (2.6), is performed numerically.

2.2.3 Interatomic Potentials

Interatomic potentials, describe the energy of a system as a sum of energies owed to interactions between the constituent particles. In particular, for an N-body system, with $\{\boldsymbol{r}_n\}_{n=1}^{N}$ being the particle positions, it is assumed that the total potential energy of the system may be expressed in the form

$$V = \sum_{k=1}^{N} \sum_{n_1,\ldots,n_k} V_k(\boldsymbol{r}_{n_1}, \ldots, \boldsymbol{r}_{n_k}) \tag{2.7}$$

where the subscript k of V_k indicates the energy due to a $k-$body interaction and the subscripts $n_1, \ldots, n_k \in \{1, \ldots, N\}$, iterate over particle positions.

Lennard-Jones potential

The Lennard-Jones potential, was first introduced by John Lennard-Jones in 1924 [21], to model the potential energy of the interaction of two neutral atoms or molecules, as a function of their separation distance r. The model consists of a long range attractive term $\sim r^{-6}$ describing a London dispersion force (a van der Waals force) and a shorter range repulsive term $\sim r^{-12}$ which mimics a Pauli repulsion [22]. The exponent of the repulsive force was chosen to offer computational efficiency when evaluating, but it is not uniquely defined [22]. The analytical expression for this choice of repulsive force is

$$V_{\mathrm{LJ}}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - 2\left(\frac{\sigma}{r}\right)^{6}\right], \tag{2.8}$$

where ϵ is an energy scale (the depth of the potential) and σ, also called *van der Waals radius*, is the length scale of the well. The profile of the Lennard-Jones potential is illustrated in Fig 2.5.

The total energy felt by a particle at $\boldsymbol{r}$ due to an $N-$body system with

positions $\{\boldsymbol{r}_n\}$ is given by

$$V_{\mathrm{TOT}}(\boldsymbol{r}) = \sum_{n}^{N} 4\epsilon \left[\left(\frac{\sigma}{\|\boldsymbol{r} - \boldsymbol{r}_n\|} \right)^{12} - 2 \left(\frac{\sigma}{\|\boldsymbol{r} - \boldsymbol{r}_n\|} \right)^{6} \right] \tag{2.9}$$

It should be noted that when using periodic boundary conditions, it is customary to truncate the Lennard-Jones potential at a cut-off distance, usually of $r_c = 2.5\sigma$.

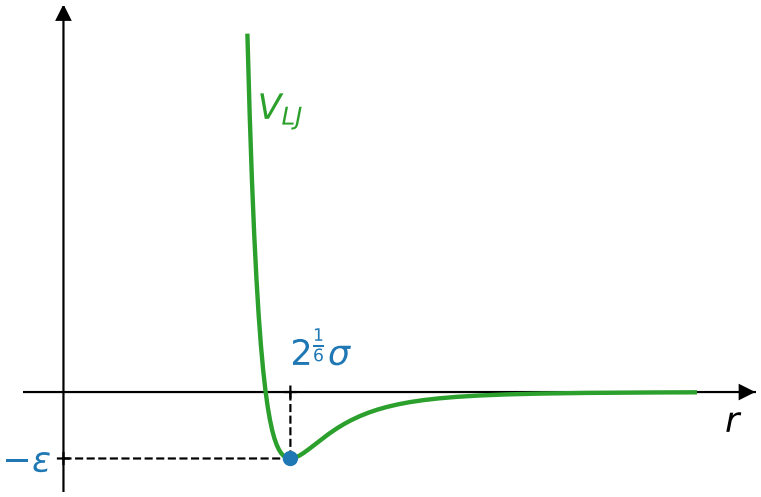


Figure 2.5: The profile of the energy profile of the Lennard-Jones potential expressed analytically by Eq. (2.8) with respect to the separation distance of two atoms or molecules.

Morse potential

In quantum mechanics potential energy curves are often reduced to quantum harmonic oscillators around their stable equilibrium points r_0, by means of truncating the Taylor expansion of the true potential around that equilibrium point after the quadratic term. Though a good approximation for molecular vibration, there are two limitations associated with the harmonic oscillator. The first being that its vibrational levels (eigenstates) are equidistant in energy but experimental evidence shows that this is not always the case for real systems. The second being that a harmonic oscillator does not predict bond breaking.

For the potential energy of a diatomic molecule, i.e. a molecule comprising of two atoms of the same chemical element, a more accurate model was proposed by P. M. Morse [23] which is widely known as the Morse potential. The analytic expression for the Morse potential as a function of the separation distance of the two atoms is given by

$$V_{\mathrm{M}}(r) = \epsilon\left(1 - e^{-A\left(\frac{r}{r_0}-1\right)}\right)^2, \tag{2.10}$$

where ϵ is the depth of the well, r_0 is the equilibrium bond distance depending on the element and A is a measure for the width of the well. For the Morse potential, the vibrational levels are not equispaced in energy.

Alternatively, one may write the Morse potential as

$$V_{\mathrm{M}}(r) = \epsilon\left(e^{-2A\left(\frac{r}{r_0}-1\right)} - 2e^{-A\left(\frac{r}{r_0}-1\right)}\right). \tag{2.11}$$

which is equivalent to Eq. (2.10) up to an additive constant. In this form one can identify a short-range repulsion term

$$\sim \exp\big\{-2A\big(r/r_0 - 1\big)\big\} \tag{2.12}$$

and a longer-range attractive term

$$\sim \exp\big\{-A\big(r/r_0 - 1\big)\big\} \tag{2.13}$$

similarly to the Lennard-Jones model Eq. (2.8).

It is also important to note that the Morse potential is a good approximation for a broader domain of r and not only in the neigbourhood of r_0 as it was the case for the harmonic oscillator.

The energy profile of the Morse potential for a diatomic atom is sketched in Fig. 2.6.

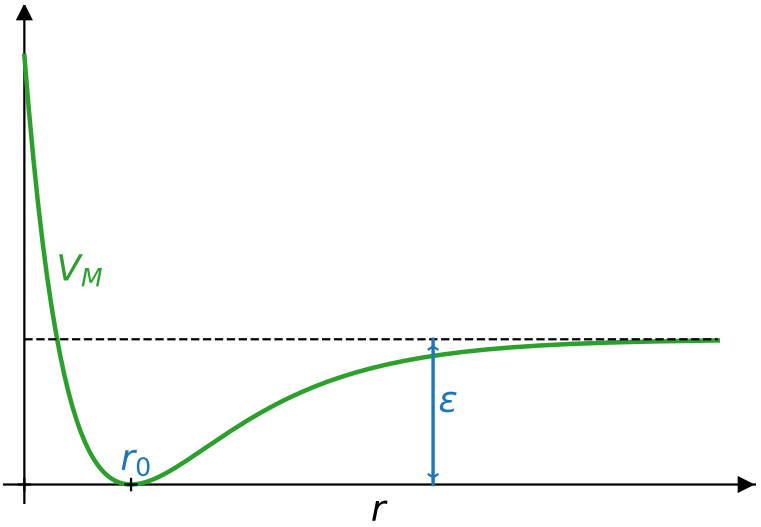


Figure 2.6: The form of the energy profile of the Morse potential describing the interaction of a diatomic molecule and described analytically by Eq. (2.10) as a function of the separation distance of two atoms comprising the molecule.

Embedded-Atom Model interatomic potential

The main disadvantage of simple pair potentials is their inadequacy to reproduce the full elastic constants of metallic systems. However, they have been and are still being used to model small elastic distortions, but they are not suitable for the modelling of big changes in bonding like in the case of surface bonding. This is because pair-potentials assume that each interatomic bond [24, 25] is independent of the rest in a system. Thus began the development of *many-body empirical models* to overcome these issues.

One such many-body interatomic potential model which we will come across in this thesis is the Embedded-Atom Model or Embedded-Atom Method (EAM). The model was proposed by Murray Daw and Mike Baskes [26, 27] but also independently by Finnis and Sinclair [28] and Gupta [29] and describes the total energy of a metallic system. Each atom in the system is considered to be embedded in the electron density of its neighbouring atoms. Then, the total energy is taken to be the sum of pair potential terms between atoms (including Coulomb repulsions between the nuclei) and the embedding energy describing the bonding of each atom with its neighbours [1, 25]. In general the

energy of any atom i in the system is expressed by

$$V_i = F\left(\sum_{j\neq i}\rho(r_{ij})\right) + \frac{1}{2}\sum_{j\neq i}\phi(r_{ij}), \tag{2.14}$$

where F is the embedding energy required to locate atom i in the electron density of its neighbours, ρ is the electron density of atom j located at a distance r_{ij} from atom i, $\phi(r_{ij})$ is the pair potential describing the interaction of atoms i with j separated by a distance r_{ij} and the summations are taken over all neighbouring atoms j of i, often taken so that the separation distance r_{ij} is less than a cut-off distance r_{cut}. In the first term, the linear interpolation of the electron densities that makes up the background electron density in which atom i is embedded, is a reasonable approximation for metallic systems [25, 30]. Then the total energy for the bulk is simply the superposition of all energies

$$V_{\text{EAM}} = \sum_i V_i. \tag{2.15}$$

The embedding energy F is defined uniquely when ρ and ϕ are determined, and this is achieved by fitting F to experimental results or DFT calculations (see Sec. 2.2.4 below) while imposing that the cohesive energy of the metal in terms of the lattice constant is consistent [30].

As pointed out by Ercolessi and Adams [31], EAM potentials are highly sensitive to the fitting database that was used, in that they can accurately simulate configurations similar to the ones included in the fitting database, but underperform otherwise. However their simple formalism and numerical efficiency justifies why many scientists prefer to use them in comparison to other models. In fact, the fitted data are as significant as the approximation used for the electronic structure effects, which lead Ercolessi and Adams to the development of the force-matching method [31].

The EAM implementation of Daw and Baskes [26, 27], was valid for FCC crystal structures, but not for BCC, Hexagonal Close Packed (HCP) and non-metallic structures [25, 32]. Other developments of the EAM potential, such as the Finnis-Sinclair [28] and the extension by Ackland and Thetford [33] have been successful in treating BCC configurations. In particular, the Finnis-

Sinclair and Ackland and Thetford potentials are notable examples of EAM models for W. However, for the treatment of W in this thesis we will be using the Marinica *et al.* description [1], which uses the force-matching method of [31] to parametrise the EAM potential on the same basis as it has been previously used by Mendelev et al [34, 35, 36] and Proville et al [37] for iron. The reason for this choice is the fact that the four Marinica EAM potentials give a better prediction of the non-degenerate core structure of a screw dislocation, which is a system of interest to us (see Sec. 2.3.2). In addition to Eq. (2.14), the model assumes that ϕ, ρ and F take the forms

$$\phi(r) = \sum_{i=1}^{n^\phi} a_i^\phi (\delta_i^\phi - r)^3 H(\delta_i^\phi - r), \tag{2.16}$$

$$\rho(r) = \sum_{i=1}^{n^\rho} a_i^\rho (\delta_i^\rho - r)^3 H(\delta_i^\rho - r), \tag{2.17}$$

$$F(\bar{\rho}) = a_1^F \sqrt{\bar{\rho}} + a_2^F \bar{\rho}^2, \tag{2.18}$$

where

$$H(r) = \begin{cases} 1 & \text{if } r \geq 0 \\ 0 & \text{if } r < 0 \end{cases}, \tag{2.19}$$

is the Heaviside step function, $a_i^\phi, a_i^\rho, a_i^F \in \mathbb{R}$ and the $\delta_i^\phi, \delta_i^\rho, \delta_i^F \in \mathbb{R}$ are cubic spline knots responsible for fitting the data. The dataset used for fitting includes (i) experimental values for lattice parameters, cohesive energies and elastic constants, (ii) formation energies of point defects for W computed with *ab initio* methods and (iii) atomic forces when a configuration is in liquid state, also calculated with *ab initio* methods. The four classes of potentials constructed, use different error measures to fit the data at the liquid state. As a result, the four classes give different screw dislocation energy barriers for a straight dislocation to migrate when zero stress is applied and in particular, the EAM4 potential gave a barrier which agrees best with the DFT results (see Sec. 2.2.4 for a description of DFT).

2.2.4 Density Functional Theory

We deviate our attention from the classical framework to introduce a quantum mechanical treatment of configuration systems, which allows the study of finer scale details (on an atomistic scale) to be investigated. We are referring to the Density Functional Theory (DFT) which has been mentioned earlier in Sec. 2.2.3. Inevitably, a quantum mechanical model would be far more computationally costly than a simulation on the molecular scale, a compromise to the finer detail accuracy offered by such a model.

In the quantum mechanical framework, the state of an N-body system is described wholly by the wavefunction Ψ_{tot} which is an anti-symmetric function of the electronic coordinates $\{\boldsymbol{r}_i\}$ and obeys the many-body Schrödinger equation. Methods which aim to solve the many-body system are being referred to as *ab initio* ($\approx$ from first principles). Routinely one starts from the Born-Openheimer approximation [38] before attempting to solve the many-body Schrödinger equation. According to this, the motion of the electrons in a molecule may be treated separately from that of the nuclei and the wavefunction of the molecule decouples like

$$\Psi_{\text{tot}} = \Psi_{\text{nuclei}}\, \Psi_{\text{electron}}, \tag{2.20}$$

on the grounds that the nuclei are much more massive compared to the electrons resulting in a large time scale separation in their motion. Subsequently the electrons are assumed to have relaxed to their ground state almost instantaneously and the energy of the nuclei corresponds to that of the electrons being in their ground state. The resulting Schrödinger equation takes the form

$$\hat{H}|\Psi_{\text{tot}}\rangle = E|\Psi_{\text{tot}}\rangle, \tag{2.21}$$

where E is the eigenvalue of ψ_{tot} and the hamiltonian $\hat{H}$ in atomic units is

given by

$$\hat{H} = \underbrace{-\sum_i \frac{1}{2}\nabla^2_{\boldsymbol{r}_i}}_{\substack{\text{electronic} \\ \text{kinetic energy}}} + \underbrace{\sum_i \sum_{j<i} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}}_{\text{electron-electron repulsion}} \underbrace{-\sum_I \frac{1}{2m_I}\nabla^2_{\boldsymbol{R}_I}}_{\text{ionic kinetic energy}} + \underbrace{\sum_I \sum_{J<I} \frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|}}_{\text{nucleon-nucleon repulsion}} \underbrace{-\sum_I \sum_i \frac{Z_I}{|\boldsymbol{R}_I - \boldsymbol{r}_j|}}_{\text{electron-nucleon attraction}}, \tag{2.22}$$

where the indices i, j sum over electrons and I, J sum over nuclei, $\{\boldsymbol{R}_I\}$ are the nucleon positions, m_I is the mass of nucleus I and Z_I is its atomic number.

A quantum mechanical approach for modelling $N-$body systems is that of DFT, which allows electronic structure calculations. It was developed by Hohenberg, and Kohn and Sham [39, 40] and it is based on the principle that the energy of the system may be represented in terms of a functional of the electron density, which in its turn is a function of spatial coordinates.

From [39] for an interacting inhomogeneous electron gas comprising of N electrons, the static external potential $v(\boldsymbol{r})$ in which it is immersed, is a unique functional of its wavefunction density

$$n(\boldsymbol{r}) = N \int \mathrm{d}\boldsymbol{r}_2 \mathrm{d}\boldsymbol{r}_3 \ldots \mathrm{d}\boldsymbol{r}_N \langle \Psi_{\text{tot}}(\boldsymbol{r}, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) | \Psi_{\text{tot}}(\boldsymbol{r}, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) \rangle, \tag{2.23}$$

up to an additive constant. Moreover, the ground state energy of the gas is also a unique functional of $n(\boldsymbol{r})$ and in fact it is the minimum of the expression

$$E = \underbrace{\int v(\boldsymbol{r}) n(\boldsymbol{r}) \mathrm{d}\boldsymbol{r}}_{\text{energy due to external potential}} + \underbrace{\frac{1}{2} \iint \frac{n(\boldsymbol{r}) n(\boldsymbol{r})'}{|\boldsymbol{r} - \boldsymbol{r}'|} \mathrm{d}\boldsymbol{r} \mathrm{d}\boldsymbol{r}'}_{\text{electron-electron Coulomb energy}} + G[n], \tag{2.24}$$

over $n(\boldsymbol{r})$, subject to

$$\begin{cases} \int n(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} = N \\ n(\boldsymbol{r}) \geq 0, \ \boldsymbol{r} \in \mathbb{R}^3 \end{cases}, \tag{2.25}$$

where $G[n]$ is a universal functional of the density. Kohn and Sham [39] postulate that it is sufficient to write $G[n]$ as the sum of the kinetic energy $T[n]$ of a system with non-interacting electrons corresponding to electron density $n(\boldsymbol{r})$ (recall, the Born-Oppenheimer approximation (2.20) and Eq. (2.22) from earlier) plus an exchange-correlation energy of an interacting system of density $n(\boldsymbol{r})$ which should be approximated. For now we will assume the simplest such approximation, the Local-Density Approximation (LDA)

$$E_{\mathrm{xc}}[n] = \int n(\boldsymbol{r})\epsilon_{\mathrm{xc}}(n(\boldsymbol{r}))\mathrm{d}\boldsymbol{r}, \tag{2.26}$$

for an electron exchange-correlation energy density $\epsilon_{\mathrm{xc}}(n(\boldsymbol{r}))$ of a homogeneous electron gas, which assumes $n(\boldsymbol{r})$ varies slowly.

In [40], it is shown that solving the constrained optimisation problem may by approximated by solving the N one-particle Schrödinger equations for a system of N non-interacting electrons moving in an effective potential

$$v(\boldsymbol{r}) + \int \frac{n(\boldsymbol{r})'}{|\boldsymbol{r} - \boldsymbol{r}'|}\mathrm{d}\boldsymbol{r'} + \mu_{\mathrm{xc}}(n), \tag{2.27}$$

where

$$\mu_{\mathrm{xc}}(n) = \frac{\mathrm{d}}{\mathrm{d}n}(n\epsilon_{\mathrm{xc}}(n)). \tag{2.28}$$

Under the above assumption, there exists an orthonormal, anti-symmetric, set of wavefunctions, $\{|\phi_i\rangle\}_{i=1}^N$, each $|\phi_i\rangle$ corresponding to one of the N non-interacting electrons. The N decoupled one-particle Schrödinger equations are

$$\left[-\frac{1}{2}\nabla^2 + v(\boldsymbol{r}) + \int \frac{n(\boldsymbol{r})'}{|\boldsymbol{r} - \boldsymbol{r}'|}\mathrm{d}\boldsymbol{r'} + \mu_{\mathrm{xc}}(n) \right] |\phi_i\rangle = \epsilon_i |\phi_i\rangle, \tag{2.29}$$

and we refer to them in literature as the Kohn-Sham equations. Solving with respect to the $\{|\phi_i\rangle\}_{i=1}^N$, one may obtain the electron desnity from

$$n(\boldsymbol{r}) = \sum_{i=1}^{N} |\phi_i(\boldsymbol{r})|^2. \tag{2.30}$$

To achieve self-consistency, Kohn and Sham use an iterative procedure: starting from an initial guess for $n(\boldsymbol{r})$, the values of $v(\boldsymbol{r})$, $\int n(\boldsymbol{r})'(|\boldsymbol{r} - \boldsymbol{r}'|)^{-1}\mathrm{d}\boldsymbol{r'}$

and $\mu_{\rm xc}(n)$ are evaluated and solving the Kohn-Sham equations (2.29) together with Eq. (2.30) leads to an updated $n(\boldsymbol{r})$. The iteration continues until self-consistency is achieved. Thereafter, all other observables of the ground state can be evaluated from $n(\boldsymbol{r})$.

The above procedure assumed knowledge of the exchange-correlation energy functional and exact expressions exist only in the case of a free electron gas which was the case in [40]. Generalising DFT for any system dictates the need for an approximation, the simplest case being LDA which we have already seen in Eq. (2.26). A much improved model which gives suitable approximations even in cases where $n(\boldsymbol{r})$ varies rapidly is the Generalised Gradient Approximations (GGA) which as the name suggests, dependents on the gradient of $n(\boldsymbol{r})$

$$E_{\rm xc}^{\rm GGA}[n] = \int \epsilon_{\rm xc}(n(\boldsymbol{r}), \nabla n(\boldsymbol{r}))\mathrm{d}\boldsymbol{r}, \tag{2.31}$$

where $\epsilon_{\rm xc}$ is now a function of n and ∇n.

Parameterisation of both LDA and GGA models is performed by fitting the respective models to Quantum Monte Carlo calculations. The matter of parameterising the GGA functional is a topic of extensive dialogue in the community, see for example [41]. For the purposes of DFT calculations in this thesis, the Perdew-Burke-Ernzerhof functional (PBE) [42] is used, which offers a simple form but captures the most important features of previous models such as [41].

Finally, we look at generating a basis set for the $\{|\phi_i\rangle\}_{i=1}^{N}$. In fact, there are many possible choices for defining a basis set, ranging from linear combinations of atomic orbitals (LCAO) [43] popular amongst the quantum chemistry community, real space grids, wavelets and plane waves, the latter being favoured by the solid-state community and also being the family of our choice. In particular, I will be using the CASTEP package [44].

For a periodic system, from Bloch's theorem [45] the wavefunction of a single electron in a crystal is spanned by the Bloch waves $u_{n,\boldsymbol{k}}$ which are periodic functions having the same periodicity as the system. Thus, in the

plane wave representation, each orbital is represented in the Fourier space by

$$\phi_{n,\boldsymbol{k}}(\boldsymbol{r}) = e^{n\boldsymbol{k}\cdot\boldsymbol{r}} u_{n,\boldsymbol{k}}(\boldsymbol{r}). \tag{2.32}$$

In the expression of Eq. (2.32), $\boldsymbol{k}$ is the wave vector, n is called the band index and it distinguishes between different Bloch waves corresponding to the same $\boldsymbol{k}$. We can write

$$\phi_{n,\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} c_{n,\boldsymbol{k}+\boldsymbol{G}} e^{i(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}} \tag{2.33}$$

where the $\boldsymbol{G}$ are reciprocal lattice vectors of the periodic system (the primitive cell of the reciprocal space is refered to as the first Brillouin zone) and the $c_{n,\boldsymbol{k}+\boldsymbol{G}}$ are the coefficients for the n^{th} band. This sum is usually truncated by some energy cutoff which is such as to not change the total energy of the system. It is important to note that one calculates the $u_{n,\boldsymbol{k}}$ only at a selected finite number of wave vectors $\boldsymbol{k}$ and the method of selecting such is important not only for computational efficiency but also to produce the correct properties for the system. The Monkhorst-Pack method [46] for selecting an appropriate grid of $\boldsymbol{k}$ vectors, thereafter called $k-$points, is used in my calculations. The Monkhorst-Pack method generates an evenly spaced rectangular grid.

2.3 Point defects

To demonstrate the effectiveness of our proposed methods we will look at a variety of examples. These include transitions in the Muëller-Brown potential landscape which was studied in Sec. 2.2.1, as well as transitions in crystal systems owed to the presence of *point defects* in the system. By defects, we mean distortions of the regular pattern of the lattice caused by say atoms of the crystal that are at irregular places of the lattice, or atoms that are missing. For a point defect we imply that the defect occurs at a single point of the lattice. In this section, we introduce the systems that we will be studying in the thesis.

2.3.1 Vacancy

A vacancy is an empty space in a lattice, which would normally have been occupied by an atom. In fact vacancies occur in crystals naturally, as the temperature of the system rises. The number of vacancies N_{V} formed in a crystal due to thermal vibrations, follows an exponential rule

$$N_{\mathrm{v}} = N_{\mathrm{A}} \exp\left(\frac{-Q_{\mathrm{v}}}{k_{\mathrm{B}}T}\right), \tag{2.34}$$

where N_{A} is the number of atoms in the system, Q_{v} is the energy needed to form a vacancy, k_{B} is the Boltzmann constant and T is the absolute temperature. But vacancies may also be formed for many other reasons, for example because of residual tensile stress in the crystal.

For our examples we look at vacancies occurring in an artificial 2D triangular lattice and a 3D supercell of a crystal structure. The vacancy is placed at the centre of the cell. To model interactions in the 2D triangular lattice, the Lennard-Jones potential (Sec. 2.2.3) is used, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$ so that the equilibrium bond length is 1.0. The system is made up of 60 atoms and periodic boundary conditions are imposed in the x and y directions. The transition of the vacancy migrating in the y direction by one lattice spacing is considered. For the 3D supercell, we will study three different cases:

1. A 107-atom Cu supercell of FCC structure in a fixed cell of periodic boundary conditions. Interactions are modelled by the Morse potential with parameters $A = 4.0$, $\epsilon = 1.0$ and nearest neighbour distance $r_0 = 2.55$ Å. The implementation used in our tests is available in the JuLIP.jl package [47].

2. A 53-atom W BCC supercell in a fixed cell and periodic conditions on the boundaries. The system is depicted in Fig. 2.7. The interactions in this case were modelled with one of

   (a) the EAM4 potential proposed in [1] and reviewed in Sec. 2.2.3. This was implemented in the JuLIP.jl package [47].

(b) a DFT model (see Sec. 2.2.4) as implemented in the CASTEP [44] software, with the exchange correlation functional approximated by the PBE GGA [42], a planewave energy cut-off of 500 eV and a $2 \times 2 \times 2$ Monkhorst-Pack grid to sample the Brillouin zone. A comparison of convergence behaviour obtained with a $3 \times 3 \times 3$ $k-$point grid was carried out which showed that the use of the $2 \times 2 \times 2$ $k-$point grid is sufficient.

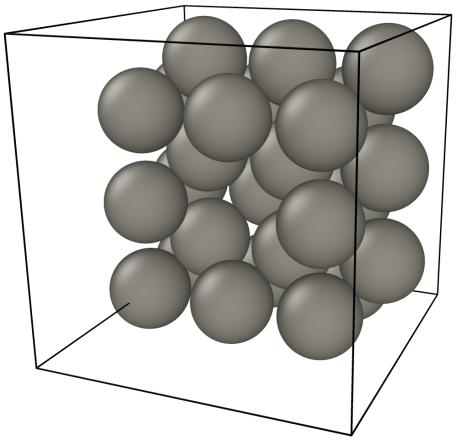


Figure 2.7: A slice of a 53-atom W BCC supercell in a fixed cell containing a vacancy in the centre.

2.3.2 Screw dislocation

There are two types of dislocations found in crystals, edge and screw dislocations and typically real materials exhibit dislocations of mixed characteristics. An edge dislocation may be visualised, by considering an extra half-plane of atoms midway through two regular planes of atoms in a system, whereas a screw dislocation may be visualised if one considers a cut along parallel planes and slipping the halves across the cut by one lattice vector, so that the shifted halves 'complete' the lattices of the unshifted ones.

Dislocations are described by a *line direction* tracing along the bottom of the extra half plane and the *Burgers vector* which gives the direction and magnitude of the distortion of the lattice. For an edge dislocation, the line

direction is normal to the Burgers vector while in a screw dislocation the two are parallel.

In our examples, we focus on the screw dislocation case [48]. We consider a $\frac{1}{2}\langle 111\rangle$ screw dislocation in a W BCC structure. In the transition of interest, the dislocation advances by one glide step. Two systems sizes are considered. In the first case, the 562-atom system is confined in a cylinder of radius equal to 20 Å and surrounded by an 11 Å cylindrical shell of clamped atoms (equal to 2 times the cutoff radius) and in the second case, the 1489-atom system is confined in a cylinder of radius 40 Å and surrounded once more by an 11 Å cylindrical shell of clamped atoms. Periodic boundary conditions are imposed along the dislocation line (z) direction. The latter is depicted in Fig. 2.8. To model interactions between atoms, we use two different models, in the first case an EAM4 potential (see Sec. 2.2.3) is used, whereas in the second case, a hybrid model is used, combining an EAM2 potential and an EAM4 potential. The Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) is used for the implementation of these potentials [49].

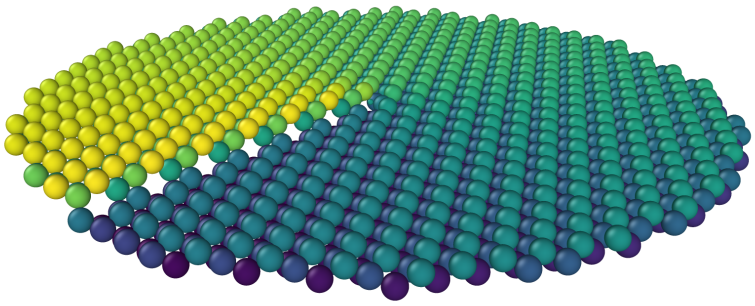


Figure 2.8: A $\frac{1}{2}\langle 111\rangle$ screw dislocation in a 1489-atom W BCC structure. Particles are coloured according to their position on the z axis.

2.4 Transition state theory

Let $\boldsymbol{x} \in \mathbb{R}^M, M \in \mathbb{N}$, be a state, or configuration, of the dynamical system in question. We denote by $V(\boldsymbol{x})$ the potential energy of $\boldsymbol{x}$ and assume that V is twice differentiable and that it has at least two local minima, $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$. These are characterised as *metastable* states as the system spends a long time

in their vicinity, before it escapes their 'pull of attraction'. Suppose that the system is at $\boldsymbol{x}_{\mathrm{A}}$. In rare occasions, the system succeeds to overcome the pull of attraction of $\boldsymbol{x}_{\mathrm{A}}$ and might transition to $\boldsymbol{x}_{\mathrm{B}}$. It is also possible that the system will return to $\boldsymbol{x}_{\mathrm{A}}$ or transition to a different metastable state if one such exists.

Such thermally activated transitions between metastable states are very rarely observed in molecular dynamics. This is because, the time spent by the system in a metastable state, before it escapes, is much longer than the typical timescale of simulation. Molecular simulations run on a timescale comparable to that of molecular vibrations ($\sim 10^{-12}$s) [50] whereas the underlying rare events take place within a timeframe of several orders of magnitude more. Their long timescale is owed to the great separation of scales of typical energy barriers of potential wells $\Delta E \sim 0.5$eV and thermal effects $\mathcal{O}(k_B T) \sim 0.025$eV. In the notation, k_{B} is the Boltzmann constant and T is the temperature of the system.

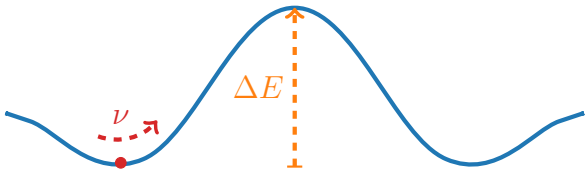


Figure 2.9: Schematic of an energy profile of a typical thermally activated transition. The energy barrier ΔE is indicated on the diagram.

However, knowledge of the transition rate ν enables the simulation of the transition on the mesoscale using, for example, the kinetic Monte Carlo method [5]. The evaluation of the transition rate to leading order is possible with Arrhenius' law [6, 7, 4],

$$\nu \sim \nu_0 \exp\left({}^{-\Delta E}/_{k_{\mathrm{B}} T}\right), \tag{2.35}$$

where the attempt rate ν_0 may be estimated using Eyring's heuristic derivation [7], or approximated with Harmonic Transition State Theory [8]. In this thesis, we are not pursuing this problem, but we included the above brief for motivation of the problem of interest.

As the transition rate scales exponentially with ΔE, improving the

accuracy and robustness of the energy barrier estimate leads to significant improvements in accuracy of the transition rate.

To find the energy barrier ΔE one should identify the saddle between $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ [51]. The following assumption is made: $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ are separated by a single saddle point $\boldsymbol{x}_{\mathrm{S}}$ of Morse index 1, so that there is a unique direction of steepest descent at the saddle $\boldsymbol{x}_{\mathrm{S}}$ [52], where $\boldsymbol{x}_{\mathrm{S}}$ is called the *transition state* which we aim to identify.

The computational evaluation of the Minimum Energy Path (MEP) [53] of the transition is a familiar technique used to find the transition state assuming that both $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{B}}$ are known, but other techniques exists for identifying the saddle point directly which assume only knowledge of $\boldsymbol{x}_{\mathrm{A}}$ [51].

An MEP of the transition from $\boldsymbol{x}_{\mathrm{A}}$ to $\boldsymbol{x}_{\mathrm{B}}$, is defined as the intrinsically parametrised path $\boldsymbol{x}^*(s),\ s \in [0,1]$, satisfying

$$\begin{cases} \nabla^{\perp} V(\boldsymbol{x}^*) \equiv \boldsymbol{0}, \\ \boldsymbol{x}^*(0) = \boldsymbol{x}_{\mathrm{A}}, \\ \boldsymbol{x}^*(1) = \boldsymbol{x}_{\mathrm{B}}, \end{cases} \tag{2.36}$$

where

$$\nabla^{\perp} V(\boldsymbol{x}) = \left(\mathbf{I} - \frac{\boldsymbol{x}'}{\|\boldsymbol{x}'\|} \otimes \frac{\boldsymbol{x}'}{\|\boldsymbol{x}'\|} \right) \nabla V(\boldsymbol{x}), \tag{2.37}$$

and where $\boldsymbol{x}' = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s}$ is the tangent along the path. Strictly speaking, $\nabla^{\perp} V$ depends on $\boldsymbol{x}'$ as well as $\boldsymbol{x}$ but for simplicity of notation we will only write $\nabla^{\perp} V(\boldsymbol{x})$. Figure 2.10 shows the contours of the Müller-Brown potential (described in Sec. 2.2.1) and an MEP of a transition of a system in the Müller-Brown potential landscape from a reactant state $\boldsymbol{x}_{\mathrm{A}}$ to a product state $\boldsymbol{x}_{\mathrm{B}}$. In this case the assumption that there is a unique saddle separating the local minima is not satisfied, but one may consider the transition as having two separate phases, a transition from $\boldsymbol{x}_{\mathrm{A}}$ to $\boldsymbol{x}_{\mathrm{C}}$ followed by another one from $\boldsymbol{x}_{\mathrm{C}}$ to $\boldsymbol{x}_{\mathrm{B}}$.

As stated in Chapter 1, the objective of this thesis is to improve the convergence speed of saddle point search methods, particularly MEP search methods in complicated geometries.

Before we look at the current methods for saddle point finding, we

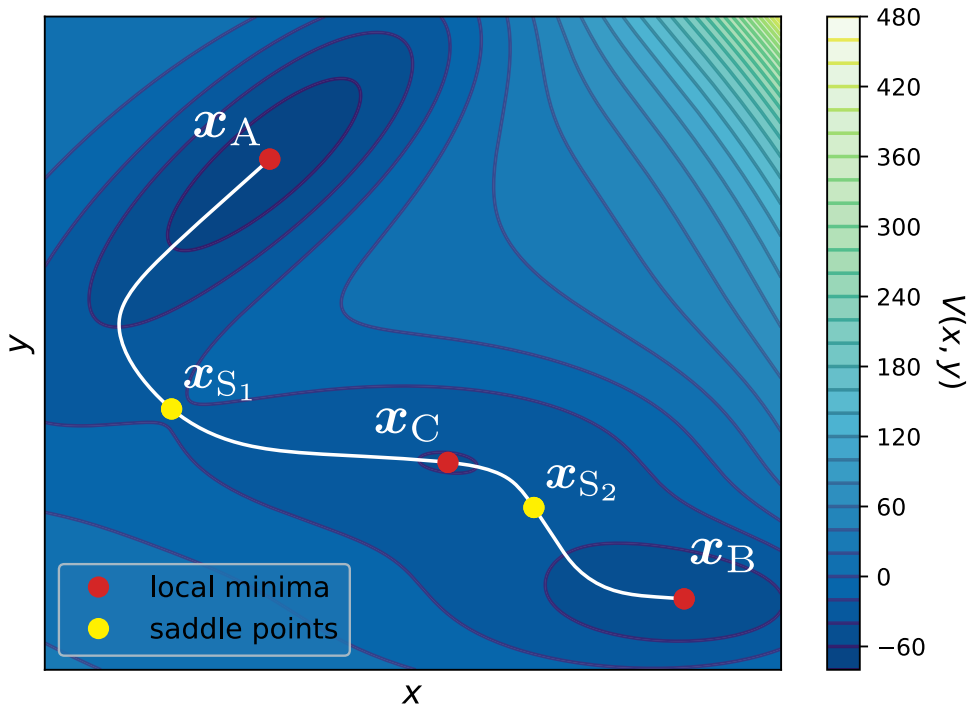


Figure 2.10: Contours of the Müller-Brown potential. The MEP of the transition from state $\boldsymbol{x}_\mathrm{A}$ to state $\boldsymbol{x}_\mathrm{B}$ is shown in white. The transition has two saddle points, $\boldsymbol{x}_{\mathrm{S}_1}$ and $\boldsymbol{x}_{\mathrm{S}_2}$ marked in yellow along the MEP. The transition may be thought of as two separate transitions: from $\boldsymbol{x}_\mathrm{A}$ to $\boldsymbol{x}_\mathrm{C}$ and another one from $\boldsymbol{x}_\mathrm{C}$ to $\boldsymbol{x}_\mathrm{B}$ to satisfy the assumptions made in our analysis.

introduce the concept of *geometry optimisation*.

2.5 Geometry Optimisation

Classical systems, prefer to be in a state which minimises their energy. Therefore, if the system in question is in state $\boldsymbol{x}(t_0)$ at time t_0, we are interested in finding the state to which the system relaxes to as $t \to \infty$. This is the problem addressed by *geometry optimisation* or *energy optimisation*. Numerous algorithms exist relating to this task. We take a closer look to a few of these algorithms which are relevant to us. These are iterative schemes and with one exception all algorithms addressed come from the general framework of optimisation in numerical analysis: starting with an initial guess, $\boldsymbol{x}^0$, they proceed iteratively, taking incremental steps along a *search direction* until convergence

to the minimum.

2.5.1 Steepest Descent

Steepest descent or *gradient descent* is the prototypical method to minimisation. It only requires V to be differentiable once. Let $\boldsymbol{x}^k$ be the state after k minimisation steps, steepest descent advances by an amount α^k in the direction, $-\nabla V(\boldsymbol{x}^k)$, in which V decreases most rapidly [15]. This can be seen by considering the first order Taylor expansion of V near $\boldsymbol{x}$,

$$V(\boldsymbol{x} + \epsilon \boldsymbol{p}) = V(\boldsymbol{x}) + \epsilon \boldsymbol{p} \cdot \nabla V(\boldsymbol{x}) + \mathcal{O}(\epsilon^2). \tag{2.38}$$

for small ϵ. The rate of decrease of V along direction $\boldsymbol{p}$ is the coefficient of the first-order term. Maximising $V(\boldsymbol{x}+\epsilon\boldsymbol{p})-V(\boldsymbol{x})$ with respect to $\boldsymbol{p}$ and requiring that $||\boldsymbol{p}|| = 1$ we find the direction of most rapid decrease to be

$$\boldsymbol{p} = -\frac{\nabla V(\boldsymbol{x})}{\|\nabla V(\boldsymbol{x})\|}. \tag{2.39}$$

The steepest descent iteration step is expressed by

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k \nabla V(\boldsymbol{x}^k). \tag{2.40}$$

Notice that steepest descent can be derived from employing Euler's method to solve the first order differential equation,

$$\dot{\boldsymbol{x}} = -\nabla V(\boldsymbol{x}). \tag{2.41}$$

Step length selection α^k is achieved by means of a *line search* [15]. A popular choice for line search is to check that α^k satisfies at every iteration k the Wolfe conditions [15], that is the Armijo condition

$$V(\boldsymbol{x}^k + \alpha^k \boldsymbol{p}^k) \leq V(\boldsymbol{x}^k) + c_1 \alpha^k \boldsymbol{p}^k \cdot \nabla V(\boldsymbol{x}^k), \tag{2.42}$$

ensuring sufficient decrease in V and a curvature condition

$$\mathbf{p}^k \cdot \nabla V(\boldsymbol{x}^k + \alpha^k \boldsymbol{p}^k) \geq c_2 \boldsymbol{p}^k \cdot \nabla V(\boldsymbol{x}^k) \tag{2.43}$$

ensuring α^k is not too small, for some constants $c_1, c_2 \in (0,1)$. For the steepest descent method, it suffices to satisfy the Armijo condition alone [15].

2.5.2 Newton's Method

Let V be twice differentiable and $\nabla\nabla V(\boldsymbol{x})$ be positive definite. The Taylor expansion of V around $\boldsymbol{x}$ up to second order terms is:

$$V(\boldsymbol{x}+\epsilon\boldsymbol{p}) = V(\boldsymbol{x}) + \epsilon\boldsymbol{p}\cdot\nabla V(\boldsymbol{x}) + \frac{1}{2}\epsilon^2\boldsymbol{p}\cdot\nabla\nabla V(\boldsymbol{x})\boldsymbol{p} + \mathcal{O}(\epsilon^3). \tag{2.44}$$

Maximising the second order approximation of $V(\boldsymbol{x}+\epsilon\boldsymbol{p}) - V(\boldsymbol{x})$ with respect to $\boldsymbol{p}$, we find

$$\boldsymbol{p} = -\big(\nabla\nabla V(\boldsymbol{x})\big)^{-1}\nabla V(\boldsymbol{x}). \tag{2.45}$$

This is known as Newton's search direction and the associated optimisation scheme updates $\boldsymbol{x}^k$ by

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k\big(\nabla\nabla V(\boldsymbol{x}^k)\big)^{-1}\nabla V(\boldsymbol{x}^k). \tag{2.46}$$

In the vicinity of the local minimum, it can be shown that Newton's method converges quadratically[1], provided that as $k \to \infty$ the natural choice for the step length is always $\alpha^k = 1$. We will revisit the convergence of Newton's method in Sec. 2.6.

Newton's method, however, suffers from the fact that inverting the Hessian, $\nabla\nabla V$, takes $\mathcal{O}(N^3)$ operations which can be very computationally costly especially for models such as DFT where it is not analytically available.

2.5.3 Limited memory BFGS

The Limited memory Broyden - Fletcher - Goldfarb - Shanno optimisation algorithm (LBFGS) [54], is a quasi-Newton algorithm, i.e. it is a method based on Newton's method which estimates the inverse of the Hessian matrix. It differentiates from the original BFGS method in that in the original algorithm

[1] quadratic convergence means that the square of the error at iteration k is proportional to the error at iteration $k+1$

a dense full matrix is used to approximate the Hessian, whereas in LBFGS, only a small number of vectors is stored, representing the approximation.

This is, once again, an iterative algorithm. An estimation of the inverse of the Hessian is found according to

$$\mathbf{H}^{k+1} = (\mathbf{I} - \rho^k \boldsymbol{s}^k \otimes \boldsymbol{y}^k)\mathbf{H}^k(\mathbf{I} - \rho^k \boldsymbol{y}^k \otimes \boldsymbol{s}^k) + \rho^k \boldsymbol{s}^k \otimes \boldsymbol{s}^k \tag{2.47}$$

where,

$$\boldsymbol{s}^k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k, \tag{2.48}$$

$$\boldsymbol{y}^k = \nabla V(\boldsymbol{x}^{k+1}) - \nabla V(\boldsymbol{x}^k) \text{ and} \tag{2.49}$$

$$\rho^k = (\boldsymbol{y}^k \cdot \boldsymbol{s}^k)^{-1}. \tag{2.50}$$

Assuming that only the last M estimates of any quantity are available, to estimate the Newton direction, two loops take place. In the first, starting from $\boldsymbol{q}_M^k = -\nabla f(\boldsymbol{x}^k)$, $\boldsymbol{q}_\ell^k$ is updated according to

$$a_\ell^k = \rho_\ell^k(\boldsymbol{s}_\ell^k \cdot \boldsymbol{q}_\ell^k) \tag{2.51}$$

$$\boldsymbol{q}_{\ell-1}^k = \boldsymbol{q}_\ell^k - a_\ell^k \boldsymbol{y}_\ell^k. \tag{2.52}$$

repeatedly until all M stored quantities have been exhausted. The subscripts ℓ index the stored quantities at iteration k. Then setting $\boldsymbol{z}_0^k = \boldsymbol{q}_0^k$, the second loop takes place,

$$b_\ell^k = \rho_\ell^k(\boldsymbol{y}_\ell^k \cdot \boldsymbol{z}_\ell^k) \tag{2.53}$$

$$\boldsymbol{z}_{\ell+1}^k = \boldsymbol{z}_\ell^k + \boldsymbol{s}_\ell^k(a_\ell^k - b_\ell^k). \tag{2.54}$$

updating $\boldsymbol{z}_\ell^k$ until all M stored quantities have been exhausted. The estimation of the Newton direction for step $k+1$ is simply $-\boldsymbol{z}_{M-1}^k$. To ensure that LBFGS updating is stable, it is necessary that line search is used, imposing the Wolfe conditions (2.42) and (2.43) [15].

2.5.4 Conjugate Gradient

The conjugate gradient method originated from the work of M. Hestenes and E. Stiefel in 1952 [55] and it is an iterative scheme dedicated to solving numerically the optimisation problem provided that the associated $\nabla\nabla V$ is symmetric and positive definite.

The conjugate gradient method proceeds by constructing a sequence of search directions which are conjugate to the direction of ∇V and to each other. The implementation specifics of this method are not important to us and they are therefore omitted, but we will address its convergence in Sec. 2.6.

2.5.5 Fast Inertial Relaxation Engine

The Fast Inertial Relaxation Engine (FIRE) [56] is an optimisation technique that was specifically developed for molecular dynamics. The method is based on the principle of damped molecular dynamics, i.e. to introduce a fictitious damping force proportional to the velocity $\sim -\dot{\boldsymbol{x}}$, to drive the system down in energy.

In particular, the system is evolved according to the following equation of motion

$$\ddot{\boldsymbol{x}}(t) = -m^{-1}\nabla V\big(\boldsymbol{x}(t)\big) - \gamma(t)|\dot{\boldsymbol{x}}(t)|\left(\frac{\dot{\boldsymbol{x}}(t)}{\|\dot{\boldsymbol{x}}(t)\|} + \frac{\nabla V\big(\boldsymbol{x}(t)\big)}{\|\nabla V\big(\boldsymbol{x}(t)\big)\|}\right) \tag{2.55}$$

for a mass m and $\gamma(t)$ chosen so that the power, calculated from $-\nabla V(\boldsymbol{x})\cdot\dot{\boldsymbol{x}}$, is positive and that the system will accelerate in a direction that is steeper than the current direction of motion.

The path traversed in energy landscape $V(\boldsymbol{x})$ as defined by the equation of motion (2.55) was characterised in [56] by its authors to be analogous to that taken by a blind skier looking for the fastest way to the valley of an unknown mountainscape.

2.5.6 Momentum Descent

There is a family of methods, dedicated to accelerating steepest descent methods, that of *momentum descent* or *accelerated gradient descent*. The idea

behind momentum descent is to introduce some form of momentum which will accelerate the dynamics. Early examples include Polyak's heavy ball method [57],

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha \nabla V(\boldsymbol{x}^k) + \beta(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}), \tag{2.56}$$

which employs the momentum term, $\beta(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$, to accelerate convergence and Nesterov's accelerated gradient descent [58] given by

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{v}^k - \alpha \nabla V(\boldsymbol{v}^k), \\ \boldsymbol{v}^{k+1} &= \boldsymbol{x}^{k+1} + \frac{k-1}{k+2}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k). \end{aligned} \tag{2.57}$$

We will address momentum descent in more detail in Sec. 5.

2.6 Ill-conditioning

For a twice-differentiable potential V, the quantity

$$\kappa(V(x)) = \|(\nabla\nabla V(x))^{-1}\|\|\nabla\nabla V(x))|, \tag{2.58}$$

satisfying $\kappa(V) \geq 1$, defines the *condition number* of V, where the choice of the norm is arbitrary. We will take $\|\cdot\|$ to be the ℓ^2-norm, in which case, the condition number reduces to

$$\kappa(V) = \frac{\sigma_{\max}(\nabla\nabla V)}{\sigma_{\min}(\nabla\nabla V)}, \tag{2.59}$$

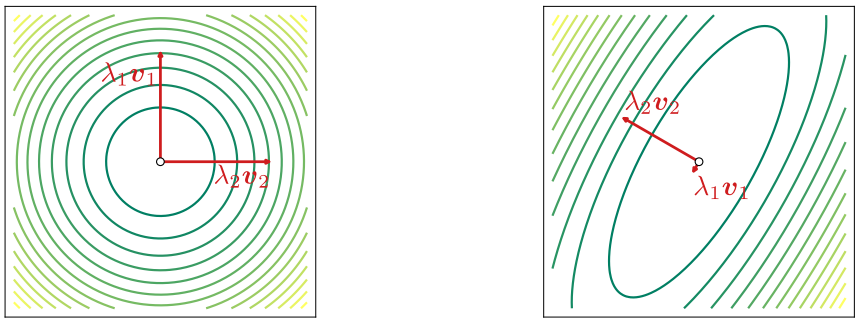
where $\sigma_{\max}$ and $\sigma_{\min}$ denote the maximum and minimum singular values of $\nabla\nabla V$. Because $\nabla\nabla V$ is symmetric, Eq. (2.59) reduces further to

$$\kappa(V) = \frac{\max_m |\lambda_m|}{\min_m |\lambda_m|}, \tag{2.60}$$

where $\{\lambda_m\}_{m=1}^M$ are the eigenvalues of $\nabla\nabla V$.

In the neighbourhood of a local minimum, $\lambda_m > 0\ \forall m = 1, \ldots, M$. We assign the relative ordering $0 < \lambda_1 \leq \cdots \leq \lambda_M$. Suppose, $\kappa \approx 1$, then, the contours of V become approximately spherical in the neighbourhood of the minimum and V is characterised as *well-conditioned*. On the other hand, as

κ gets larger, the contours of V become skew-elliptical and V is characterised as *ill-conditioned*. This is best illustrated on Fig. 2.11.



(a) well-conditioned V (b) ill-conditioned V

Figure 2.11: The diagrams show the contours of (a) a well-conditionded V near equilibrium and (b) an ill-conditioned function V near equilibrium.

To understand the implications of a small or a large condition number in geometry optimisation, we look at the ideal case where V is a strongly convex function, that is

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot \mathbf{Q}\boldsymbol{x} - \boldsymbol{d} \cdot \boldsymbol{x}, \tag{2.61}$$

where $\mathbf{Q} \in \mathbb{R}^{M \times M}$ is symmetric and positive definite, and $\boldsymbol{d} \in \mathbb{R}^M$. There is an exact step length α^k which minimises $V(\boldsymbol{x}^k - \alpha \nabla V(\boldsymbol{x}^k))$,

$$\alpha^k = \frac{V(\boldsymbol{x}^k) \cdot V(\boldsymbol{x}^k)}{V(\boldsymbol{x}^k) \cdot \mathbf{Q}V(\boldsymbol{x}^k)}. \tag{2.62}$$

The following is true for the convergence to a minimum $\boldsymbol{x}^*$ of the steepest descent method (Sec. 2.5.1) with step sizes (2.62) [15]:

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}} \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_{\mathbf{Q}}. \tag{2.63}$$

This implies that as κ increases, the ratio

$$\frac{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_{\mathbf{Q}}}{\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}}} \approx 1 \tag{2.64}$$

and the method converges very slowly.

For the conjugate gradient method introduced in Sec. 2.5.4 and for this particular V, let

$$\boldsymbol{r}^k = \boldsymbol{d} - \mathbf{Q}\boldsymbol{x}^k, \tag{2.65}$$

be the residual error at iteration k, the conjugate search directions $\boldsymbol{p}^k$ are constructed according to

$$\boldsymbol{p}^0 = \boldsymbol{r}^0, \tag{2.66}$$

$$\boldsymbol{p}^k = \boldsymbol{r}^k - \sum_{i<k} \left(\frac{\boldsymbol{p}^i \cdot \mathbf{Q}\boldsymbol{r}^k}{\boldsymbol{p}^i \cdot \mathbf{Q}\boldsymbol{p}^i} \right) \boldsymbol{p}^i, \tag{2.67}$$

and the appropriate choice for α^k is

$$\alpha^k = \frac{\boldsymbol{p}^k \cdot \boldsymbol{r}^k}{\boldsymbol{p}^k \cdot \mathbf{Q}\boldsymbol{p}^i}. \tag{2.68}$$

Then one can show that the following is true about the convergence to a minimum $\boldsymbol{x}^*$ of the conjugate gradient method.

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}} \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_{\mathbf{Q}}. \tag{2.69}$$

On the other hand, we have seen in Sec. 2.5.2 that with Newton's method, the sequence of $\boldsymbol{x}^k$ converges quadratically to $\boldsymbol{x}^*$. In particular, there is a constant $\Lambda(V)$ [15] such that,

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\| \leq \Lambda(V)\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2. \tag{2.70}$$

2.7 Cubic Spline Interpolation

In preparation for discussing the MEP finding methods, we briefly review cubic spline interpolation. Let $\{(s_n, \boldsymbol{x}_n)\}_{n=0}^N$ be a collection of $N+1$ known and distinct pairs such that $\boldsymbol{x}_n \in \mathbb{R}^M$ $\forall n$ and $s_n \in \mathbb{R}$ $\forall n$ are arranged such that

$$s_1 < \cdots < s_i < s_{i+1} < \cdots < s_N. \tag{2.71}$$

We call s_n the *knots*. We wish to find a function, $\boldsymbol{x} : [s_1, s_N] \to \mathbb{R}^M$, such that $\boldsymbol{x}(s_n) = \boldsymbol{x}_n \ \forall n$. This is known as *interpolation* of the $N+1$ pairs. In the context of the thesis we are interested in $\boldsymbol{x}$ being twice continuously differentiable in s. We describe here the method of *cubic spline interpolation*.

Cubic spline interpolation [59], constructs for each coordinate m of $(\boldsymbol{x})_m$ a piecewise polynomial function $(\boldsymbol{x}(s))_m$, using cubic polynomials to interpolate $(\boldsymbol{x}_i)_m, (\boldsymbol{x}_{i+1})_m$ in each subinterval $[s_i, s_{i+1}]$. It enforces continuity of the first and second derivatives at the knots. The remaining 2 degrees of freedom are fixed by enforcing a boundary condition. Common boundary conditions are the following:

- *natural boundary condition*: The second derivative is zero at the end points, $\boldsymbol{x}''(s_0) = \boldsymbol{x}''(s_N) = 0$.
- *not-a-knot boundary condition*: The third derivative is continuous at the end points, $\boldsymbol{x}'''(s_0) = \boldsymbol{x}'''(s_1)$ and $\boldsymbol{x}'''(s_{N-1}) = \boldsymbol{x}'''(s_N)$.
- *clamped boundary condition*: The first derivative at the end points is known.

In the present work for interpolation purposes we employ the cubic spline interpolation and we impose the "not-a-knot" boundary condition, but the methods we discuss can be readily extended to other interpolation schemes as well.

2.8 Minimum Energy Path finding: the NEB and String methods

Returning to MEP finding, we consider two notable methods, which are used extensively in the literature: the *string method* [10, 11, 12] and the *Nudged Elastic Band* (NEB) method [13, 14]. Both find the MEP by iteratively relaxing a discretised path $\boldsymbol{x}(s)$ of N images $\{\boldsymbol{x}_n\}_{n=1}^N$, until convergence to an approximate MEP is achieved. Typically, the path is evolved in the energy landscape via a steepest descent-like optimisation technique; thus, the methods may converge slowly when the potential is ill-conditioned. Such a situation

arises, for example, in large computational domains or if bonds with significant stiffness variations are present. One of the main contributions of this thesis is to propose preconditioning techniques that overcome this limitation.

From equation (2.36), notice that points along the MEP are steady state solutions of the ODEs

$$\begin{aligned} \dot{\boldsymbol{x}}_n &= -\nabla^{\perp} V(\boldsymbol{x}_n), \quad \forall n = 2, \ldots, N-1 \\ \dot{\boldsymbol{x}}_1 &= -\nabla V(\boldsymbol{x}_1), \quad \dot{\boldsymbol{x}}_N = -\nabla V(\boldsymbol{x}_N), \end{aligned} \tag{2.72}$$

where we have introduced the pseudo-temporal coordinate τ and we use the notation

$$\dot{\boldsymbol{x}} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\tau}. \tag{2.73}$$

Starting from an initial guess, we can evolve a discretised path according to the flow of Eq. (2.72) to approach asymptotically the MEP. However, images should not be evolved independently of the rest, as in this case there is nothing preventing them from converging to the nearest local minima of V. So instead, the evolution of $\boldsymbol{x}_n(\tau)$ in string and NEB is described by the system of ODEs

$$\dot{\boldsymbol{x}}_n = \pi(-\nabla^{\perp} V(\boldsymbol{x}_n) + \boldsymbol{\eta}_n). \tag{2.74}$$

For the NEB method, η describes elastic interactions between adjacent images confined in the direction along the path according to

$$\boldsymbol{\eta} = \boldsymbol{\eta}_{\mathrm{neb}} = \kappa \left(\boldsymbol{x}'' \cdot \frac{\boldsymbol{x}'}{\|\boldsymbol{x}'\|} \right) \frac{\boldsymbol{x}'}{\|\boldsymbol{x}'\|}, \tag{2.75}$$

for a spring constant κ, ensuring that in regions of higher curvature the images will feel stronger elastic attraction. The function $\pi : \boldsymbol{x} \mapsto \boldsymbol{x}$ is the identity function.

While for NEB the presence of the elastic interaction $\boldsymbol{\eta}$ enforces an approximate equidistribution of the nodes along the path, the string method uses $\pi : \mathbb{R}^M \to \mathbb{R}^M$ to reparameterise the path after each iteration and ensure that the images remain equidistant with respect to a suitable metric. The value of η is set to zero. In the continuous limit, as $N \to \infty$ a converged discretised path tends to the correct MEP, independently of the choice of the reparameterisation metric [12]. We initially use the standard ℓ^2-norm defined

by $\|\boldsymbol{x}\|^2 = \boldsymbol{x} \cdot \boldsymbol{x}$, but we will introduce a different notion of distance later on in Chapter 4.

The system (2.74) can be solved numerically with any ODE numerical integrator. Most commonly, Euler's method [11] is used, which yields an update step of the form

$$\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k + \alpha^k \left[-\nabla^{\perp} V(\boldsymbol{x}_n^k) + \boldsymbol{\eta}_n^k \right], \tag{2.76}$$

where $\boldsymbol{\eta}_n^k = \boldsymbol{\eta}((\boldsymbol{x}_n^k)', (\boldsymbol{x}_n^k)'')$ and α^k is the timestep at iteration k. For the string method only, there is an additional redistribution of the images after the update step. We follow precisely the approach described in Eq. 12 in [11], but for simplicity of presentation do not make this step explicit. The redistribution requires finding an expression for $\boldsymbol{x}(s)$ which we obtain by interpolating the positions of the images with cubic spline interpolation. The redistribution step will be revisited in more detail in Chapter 4.

The updating steps Eq. (2.76) for the string and NEB methods as well as the subsequent analysis were defined in terms of total derivatives of the path variable $\boldsymbol{x}$ (i.e. in terms of $\boldsymbol{x}'$ and $\boldsymbol{x}''$), as they are motivated from the respective laws of classical dynamics. This information is available at each iteration at negligible extra cost as we use cubic spline interpolation to find an expression for $\boldsymbol{x}(s)$[10, 13].

Many variants of the string and NEB methods exist. Three notable examples are the simplified string [11] and the improved tangent estimate [14] and climbing image [60] for NEB. The simplified string method [11] omits projecting the potential gradient on the normal directions to the path. In particular, the updating step (2.76) takes the form

$$\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k - \alpha^k \nabla V(\boldsymbol{x}_n^k), \tag{2.77}$$

with the redistribution step remaining unchanged. The improved tangent estimate for NEB in [14] leads to an improved estimate of the spring forces which helps against kink formation along the path. In particular, $\boldsymbol{\eta}_n^k$ becomes

$$\boldsymbol{\eta}_n^k = \kappa\big(\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| - \|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|\big) \frac{\boldsymbol{x}_n'}{\|\boldsymbol{x}_n'\|}. \tag{2.78}$$

In the climbing image NEB [60] after a few standard NEB evolutions, the image of highest energy $\boldsymbol{x}^k_{n_{\max}}$ is evolved according to

$$\boldsymbol{x}^{k+1}_{n_{\max}} = \boldsymbol{x}^k_{n_{\max}} - \alpha^k \left(\mathbf{I} - 2 \frac{(\boldsymbol{x}^k_{n_{\max}})'}{\|(\boldsymbol{x}^k_{n_{\max}})'\|} \otimes \frac{(\boldsymbol{x}^k_{n_{\max}})'}{\|(\boldsymbol{x}^k_{n_{\max}})'\|} \right) \nabla V(\boldsymbol{x}^k_{n_{\max}}), \tag{2.79}$$

so that it eventually converges to the saddle point. In this thesis we will be using the original NEB and string methods, given by (2.76), but our analysis may be used together with any of the string or NEB variants straightforwardly.

2.9 Dimer method

Let us now discuss a method which assumes no knowledge of the final state of the transition. The *dimer method* was developed by Henkelman and Jònsson in 1999 [9] and likewise to the string and NEB methods, it is an iterative scheme, evolving the position of a pair of images until they converge to the saddle point. An extensive analysis of the convergence of the dimer method is found in [61] where the dimer method is studied as a dynamical system.

Starting from a configuration $\boldsymbol{x}^0 \in \mathbb{R}^M$, which is either at the local minimum, or at a guess closer to the saddle point, a replica of the image is constructed by displacing $\boldsymbol{x}^0$ by a vector $L\tilde{\boldsymbol{v}}^0$, where $\tilde{\boldsymbol{v}}^0$ is a unit vector in $\mathbb{R}^M$ which is aligned with the lowest curvature mode of V, i.e the greatest in magnitude negative eigenvector of $\nabla\nabla V$. A finite difference approximation of $\nabla\nabla V$ is used to find this eigenvector, which employs only the first derivatives of the potential. Aligning $\tilde{\boldsymbol{v}}$ at each iteration, is referred to as the rotation step of the method.

We deviate from the original definition of the method described in [9, 62] and instead we use the form derived in [16, 61]. It was shown in [16], that the rotational force reduces to

$$\begin{aligned} \tilde{\boldsymbol{v}}^k &= \frac{\boldsymbol{v}^k}{\|\boldsymbol{v}^k\|}, \\ \boldsymbol{f}^k_{\mathrm{v}} &= -\big(\mathbf{I} - \tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\big) \frac{\nabla V\big(\boldsymbol{x}^k + L\tilde{\boldsymbol{v}}^k\big) - \nabla V(\boldsymbol{x}^k)}{L}. \end{aligned} \tag{2.80}$$

Assuming that there is a unique direction of steepest descent at the saddle, to

approach the saddle, $\boldsymbol{x}^k$ ascends in energy in the direction of lowest curvature mode of V, given by $\tilde{\boldsymbol{v}}^k$, but descends in energy along all directions normal to $\tilde{\boldsymbol{v}}^k$. This is realised by a step in the direction

$$\begin{aligned}\boldsymbol{f}_{\mathrm{x}}^k &= \tilde{\boldsymbol{v}}^k\big(\tilde{\boldsymbol{v}}^k \cdot \nabla V(\boldsymbol{x}^k)\big) - \Big(\nabla V(\boldsymbol{x}^k) - \tilde{\boldsymbol{v}}^k\big(\tilde{\boldsymbol{v}}^k \cdot \nabla V(\boldsymbol{x}^k)\big)\Big) \\ &\equiv -\big(\mathbf{I} - 2\tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\big)\nabla V(\boldsymbol{x}^k)\end{aligned} \tag{2.81}$$

Then the image and its replica need to move according to

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_{\mathrm{x}}^k\Big(\mathbf{I} - 2\tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\Big)\nabla V(\boldsymbol{x}^k) \tag{2.82}$$

$$\boldsymbol{v}^{k+1} = \tilde{\boldsymbol{v}}^k - \alpha_{\mathrm{v}}^k\Big(\mathbf{I} - \tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\Big)\frac{\nabla V\Big(\boldsymbol{x}^k + L\tilde{\boldsymbol{v}}^k\Big) - \nabla V(\boldsymbol{x}^k)}{L} \tag{2.83}$$

$$\tilde{\boldsymbol{v}}^{k+1} = \frac{\boldsymbol{v}^{k+1}}{\|\boldsymbol{v}^{k+1}\|} \tag{2.84}$$

A preconditioned variant of the dimer method is proposed in [16], where the evolution equations take the form

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_{\mathrm{x}}^k\Big(\mathbf{P}(\boldsymbol{x}^k)^{-1} - 2\tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\Big)\nabla V(\boldsymbol{x}^k) \tag{2.85}$$

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \alpha_{\mathrm{v}}^k\Big(\mathbf{P}(\boldsymbol{x}^k)^{-1} - \tilde{\boldsymbol{v}}^k \otimes \tilde{\boldsymbol{v}}^k\Big)\frac{\nabla V\Big(\boldsymbol{x}^k + L\tilde{\boldsymbol{v}}^k\Big) - \nabla V(\boldsymbol{x}^k)}{L} \tag{2.86}$$

$$\tilde{\boldsymbol{v}}^{k+1} = \frac{\boldsymbol{v}^{k+1}}{\|\boldsymbol{v}^{k+1}\|_{\mathrm{P}}} \tag{2.87}$$

where $\mathbf{P}^k = \mathbf{P}(\boldsymbol{x}^{(k)})$.

2.10 Thesis Objective

As mentioned earlier, because both the NEB and string methods use a projected steepest-descent direction in order to evolve a tentative path, the methods inevitably inherit its slow convergence. To surpass issues arriving from ill conditioning, our proposed method is to use the numerical analysis approach of preconditioning the ill-conditioned system under consideration. The intention is to focus on preconditioning each image along a treated path separately. For this purpose, the preconditioners available for geometry optimisation are

expected to be sufficient, but we will explore how to apply them in the new context of MEP search methods.

Beyond our proposed method it should be noted that recent advances have been made towards the same goal but via rather a different route. A machine learning surrogate Gaussian process regression [63] was developed by Garrito Torres *et al.* which surpasses the problem of poor scalability of computational cost with the number of images along an NEB path, by redefining the convergence criterion which now has a probabilistic nature. As a result the surrogate model improves the robustness of the NEB method.

Our approach to improving the robustness of the method, relies on exploiting the fact that the MEPs are the steady state solutions to an ODE and the same is true for the dimer method. There is an extensive and rigorous literature dedicated to adaptive ODE numerical solvers which we will be discussing in detail in Chapter 3 and which we will adapt and optimise for saddle point search methods.

Finally, we expand the momentum descent method seen in Sec. 2.5.6 in order to utilise the method in transition state finding.

Chapter 3

Adaptive step length selection for saddle point finding methods

In this chapter we discuss step length selection for geometry optimisation methods. We first take a look at previous techniques used and we carry on to describe an early attempt which is motivated from the literature. Finally, we propose a novel *adaptive step length selection* method, which we developed to enhance time evolution of, but not limited to, string and NEB type methods; specifically robustness and on-the-fly adaptivity. We will see that the scope of applicability of this adaptive method is not limited to double-ended saddle finding methods, but in fact it can be used as a general optimisation technique and for walker-type methods such as the dimer method.

3.1 Introduction

In 2013, Samanta *et al.* [64] proposed an optimisation based string method. Their method solves an optimisation problem for each node along the path independently. For each node, $\boldsymbol{x}_n$, the method identifies the hyperplane normal to the path and in this hyperplane, the minimiser of V is found to update $\boldsymbol{x}_n$. A mixing step is employed after the minimisation step, where the new node is selected as a linear combination of the old step and the potential minimiser. Since the scheme treats nodes independently of their neighbours, it is prone to kink formation along the path, violating the smoothness condition and potentially leading to numerical instability. Another important aspect of the

method is that it disregards projections of steepest descent directions, and instead any direction normal to the path is allowed during minimisation.

For NEB and string-type methods, the LBFGS (see Sec. 2.5.3) optimisation algorithm is used widely, for example with the NEB implementation [14] in the Atomic Simulation Environment (ASE) [3]. However, this tool bares some disadvantages in our framework which will become apparent in our later analysis. Another force based optimisation tool which is often used with the NEB method is FIRE (Sec. 2.5.5) and this is available for example in the ASE library.

3.1.1 Independent image step length selection

The proposed method in [64] does not take into consideration the projections of steepest descent directions, but instead any direction normal to the path is allowed during minimisation. We state here a different method that may be used to independently evolve images along a path. This method is motivated by the standard line search techniques introduced in Chapter 2. We use the steepest descent-like, string and NEB evolution direction from (2.76)

$$\boldsymbol{f}_n^k = -\nabla^{\perp} V(\boldsymbol{x}_n^k) + \boldsymbol{\eta}_n^k \tag{3.1}$$

as the search direction.

Recall from Sec. 2.5.1 that to ensure convergence, typical line search algorithms impose the Wolfe conditions and that for a steepest descent direction it suffices to satisfy the Armijo condition alone. This can be achieved by a backtracking algorithm [15]. The algorithm is iterative, starting with a maximum initial guess $\alpha_1 = \alpha_{\max}\ \forall n$, and terminates when the Armijo condition for each individual image,

$$V\left(\boldsymbol{x}_n + \alpha_n \boldsymbol{f}_n\right) \leq V\left(\boldsymbol{x}_n\right) + c_1 \alpha_n \boldsymbol{f}_n \cdot \nabla V\left(\boldsymbol{x}_n\right), \tag{3.2}$$

is met. At each step, we improve the tentative value of the step length, α_n, for each individual image $\boldsymbol{x}_n$ by minimising the quadratic interpolant of

$$\mathcal{V}(\alpha_n) := V\left(\boldsymbol{x}_n + \alpha_n \boldsymbol{f}_n\right). \tag{3.3}$$

Following [15], this is equivalent to updating α_n by

$$\alpha_n^{k+1} \leftarrow \frac{-\frac{1}{2}\alpha_n^k \nabla V\left(\boldsymbol{x}_n^k\right) \cdot \boldsymbol{f}_n^k}{\frac{V\left(\boldsymbol{x}_n^k + \alpha_n^k \boldsymbol{f}_n^k\right) - V\left(\boldsymbol{x}_n^k\right)}{\alpha_n^k} - \nabla V\left(\boldsymbol{x}_n^k\right) \cdot \boldsymbol{f}_n^k}. \tag{3.4}$$

As each image is advanced by a different amount α_n, the resulting path will no longer be smooth and after several steps, we expect the formation of kinks along the path preventing convergence. In similar fashion to the optimisation-based string method of [64], we need to take a smoothing step to prevent kink formation along the path. We chose to mix the step sizes along the path as opposed to across path followed by [64]. To explain; provided that the old path is sufficiently smooth, we take a weighted average of the step length α_n associated with image $\boldsymbol{x}_n$ with the step lengths of its adjacent neighbours $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_{n+1}$, α_{n-1} and α_{n+1} respectively, according to

$$\alpha_n^k = \begin{cases} \frac{1}{4}\alpha_{n-1}^k + \frac{1}{2}\alpha_n^k + \frac{1}{4}\alpha_{n+1}^k, \forall n = 2, ..N-1 \\ \frac{1}{2}\alpha_n^k + \frac{1}{2}\alpha_{n+1}^k, \text{ for } n = 1 \\ \frac{1}{2}\alpha_{n-1}^k + \frac{1}{2}\alpha_n^k, \text{ for } n = N \end{cases} . \tag{3.5}$$

Of course this heuristic approach does not guarantee that the resulting path will be smooth. In fact, from the moment the images on the path move with different step lengths the smoothness condition is lost. Moreover, on areas where the steepest descent direction changes rapidly, it is expected that the described method will be more susceptible to kink formation. Another drawback is the need for the user to specify a maximum step length $\alpha_{\max}$ which needs to be hand-tuned by the user, similarly to the fixed step of the static method.

The above considerations led us to abandon this idea for good, and motivated an alternative method which treats the path as a single vector confined in $\mathbb{R}^{nM}$ as opposed to a collection of n images in $\mathbb{R}^M$

3.2 Adaptive time stepping

We start this section by observing that neither the string nor the NEB method correspond to minimisation of an energy functional, due to the projection operators involved in their forcing term (3.1).

To motivate the analysis of this section, recall from equation (2.74) in Sec. 2.8 which we restate here,

$$\dot{\boldsymbol{x}}_n = \pi(-\nabla^{\perp} V(\boldsymbol{x}_n) + \boldsymbol{\eta}_n), \tag{2.74}$$

that the string and NEB methods can be expressed as first order ODEs and that originally Euler's method was employed to derive the evolution equations (2.76) restated below

$$\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k + \alpha^k \left[-\nabla^{\perp} V(\boldsymbol{x}_n^k) + \boldsymbol{\eta}_n^k\right]. \tag{2.76}$$

There are two caveats to the use of this simple first order scheme. The first being that, a time step small enough to ensure stability for every evolution step is required and second, this step length needs to be determined by the user; a rather tedious task. However, these issues can be bypassed, if an adaptive ODE solver is used in its place. We propose the use of the adaptive ODE solver based on Hairer *et al.* [65] to allow for some adaptivity in the step selection mechanism. For an adaptive ODE solver, the user is required to supply only an absolute and a relative tolerance, *atol* and *rtol* respectively, which control the accuracy of the solution locally. We will demonstrate that choosing these two parameters is more intuitive and more robust than choosing the step length of a fixed time stepping, *static*, method.

We emphasise here that, on some occasions, a finely tuned static method might be more optimal than the adaptive step method that we will describe, and this is fine. Our aim in this chapter is not to speed up convergence of the string and NEB methods, but rather to reduce the user's effort needed to achieve convergence, by improving robustness and adaptivity.

3.2.1 The adaptive ODE solver ode12

The following description, follows the construction of a general adaptive ODE solver from [65]. Let $\boldsymbol{x}^k$ be the approximation to the solution after step k and let α_k be the step length used to propagate the solution during the same step. Now, suppose $\boldsymbol{f}(\boldsymbol{x})$ is the driving force on $\boldsymbol{x}$. We consider a first and second order explicit Runge-Kutta pair, in *non-extrapolation* mode. That is, the lowest order approximation will be used to advance the method. In particular, we take a first order Euler step, $\boldsymbol{x}^{k+1}$, and a second order Crank-Nicolson step, $\tilde{\boldsymbol{x}}^{k+1}$, given by

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{f}(\boldsymbol{x}^k), \tag{3.6}$$

$$\tilde{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^k + \frac{\alpha_k}{2}\big\{\boldsymbol{f}(\boldsymbol{x}^k) + \boldsymbol{f}(\boldsymbol{x}^{k+1})\big\}, \tag{3.7}$$

thus $\boldsymbol{x}^{k+1}$ is the trial step to advance the method. An estimate of the local error for $\boldsymbol{x}^{k+1}$ is found by $\boldsymbol{x}^{k+1} - \tilde{\boldsymbol{x}}^{k+1}$. To test the eligibility of the chosen step length, the local error is checked against a preset *absolute tolerance*, *atol*,

$$\|\boldsymbol{x}^{k+1} - \tilde{\boldsymbol{x}}^{k+1}\|_\infty \leq atol. \tag{3.8}$$

But we need to be careful when $\boldsymbol{x}^k$ becomes of the same order as *atol*. In that case, the local error estimate being small compared to *atol* is not an indication that the method has converged. To prevent this scenario, we define the relative error: for each component we look at the local error scaled by the size of the current and trial steps, and we compare this against a *relative tolerance*, *rtol*, according to

$$\frac{|x_i^{k+1} - \tilde{x}_i^{k+1}|}{\max\{|x_i^k|, |x_i^{k+1}|\}} \leq rtol, \quad \forall i = 1, \ldots, M. \tag{3.9}$$

The two inequalities are expressed together, by

$$|x_i^{k+1} - \tilde{x}_i^{k+1}| \leq \max\{atol, rtol \times \max\{|x_i^k|, |x_i^{k+1}|\}\} \quad \forall i = 1, \ldots, M. \tag{3.10}$$

We define the quantity

$$E^{k+1} := \sup_j \left\{ \frac{|x_j^{k+1} - \tilde{x}_j^{k+1}|}{rtol \times \max\left\{\frac{atol}{rtol}, |x_j^k|, |x_j^{k+1}|\right\}} \right\}, \tag{3.11}$$

which is to be thought of as an overall measure for the error and to be compared with 1, where the index j denotes vector components. For our choice of $\boldsymbol{x}^{k+1}$ and $\tilde{\boldsymbol{x}}^{k+1}$,

$$\boldsymbol{x}^{k+1} - \tilde{\boldsymbol{x}}^{k+1} = \frac{\alpha_k}{2}\{\boldsymbol{f}(\boldsymbol{x}^k) - \boldsymbol{f}(\boldsymbol{x}^{k+1})\}. \tag{3.12}$$

Thus, the error measure reduces to

$$E^{k+1} := \frac{1}{2}\alpha_k (rtol)^{-1} \sup_j \left\{ \frac{|f_j(x^k) - f_j(x^{k+1})|}{\max\left\{\frac{atol}{rtol}, |x_j^k|, |x_j^{k+1}|\right\}} \right\}, \tag{3.13}$$

which does not require the evaluation of $\tilde{\boldsymbol{x}}^{k+1}$ explicitly.

Whether $E^{k+1} < 1$ and the new step x^{k+1} is accepted or not, we want a good estimate for the new step length. From the local error analysis of the Euler method [65], it follows that

$$E^{k+1} = C_k \alpha_k^2 + \mathcal{O}(\alpha_k^3), \tag{3.14}$$

where, C_k is a constant. The optimal step size, $\alpha_{k,\text{opt}}$, is the largest step that may be taken in the search direction $\boldsymbol{f}(\boldsymbol{x}^k)$, before violating condition (3.10). From (3.10), (3.13) and (3.14), for an optimal step we have $C_k \alpha_{k,\text{opt}}^2 \approx 1$. It follows that

$$\alpha_{k,\text{opt}} = \alpha_k \sqrt{(1/E^{k+1})}. \tag{3.15}$$

If $\boldsymbol{x}^{k+1}$ is accepted, we set $\alpha_{k+1} = \alpha_{k,\text{opt}}$ and proceed to evaluate $\boldsymbol{x}^{k+2}$. Otherwise, we set $\alpha_k = \alpha_{k,\text{opt}}$ and re-evaluate $\boldsymbol{x}^{k+1}$.

Deciding on the termination criterion is where our ODE solver differs from its predecessors. The typical goal for an ODE solver is to find an approximate solution in a finite time interval $[0, T]$. Thus the typical ODE solver runs until $\sum_k \alpha_k$ exceeds T. For our case the end goal is the identification of the equilibrium $\boldsymbol{f} \equiv 0$. Thus we would like the solver to terminate when the residual R^{k+1} falls below the set tolerance, or the preset number of iterations has been exceeded. We leave the exact definition of the residual arbitrary for now, as this will permit us to extend to the string and NEB formulation later on, but for an optimisation problem, this will typically be

$$R^{k+1} = \left\| \boldsymbol{f}(\boldsymbol{x}^k) \right\|_\infty. \tag{3.16}$$

Following the standard naming convention for adaptive ODE solvers, we call the described method, ode12, where '1' and '2' reflect the first and second order approximations $\boldsymbol{x}^{k+1}$ and $\tilde{\boldsymbol{x}}^{k+1}$ considered at each iteration.

3.2.2 The ode12 solver for saddle point finding methods

We insert a small parenthesis here to discuss the implications of applying the method to saddle point finding. First we look at the string and NEB methods and then we conclude with the consideration of the dimer method.

As mentioned earlier in Sec 3.1, we would like to treat the path as a single entity. For this reason, we adopt a different notation to represent the path, which nevertheless yields equivalent updating steps of the NEB and string methods. For a discretised path $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ we define the vector entity $\mathbf{X} \in \mathbb{R}^{N\times M}$ given by

$$\mathbf{X} := \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}, \tag{3.17}$$

and the force entity $\mathbf{F} \in \mathbf{R}^{N\times M}$ given by

$$\mathbf{F} := \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_N \end{bmatrix}, \tag{3.18}$$

where $\boldsymbol{f}_n$ was defined in (3.1).

We are now ready to apply the ODE solver. The expressions for the trial step and error meassure follow naturally and are given

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \alpha_k \boldsymbol{F}^k, \tag{3.19}$$

and

$$E^{k+1} = \frac{1}{2}\alpha_k (rtol)^{-1} \sup_j \left\{ \frac{|F_j^k - F_j^{k+1}|}{\max\left\{\frac{atol}{rtol}, |X_j^k|, |X_j^{k+1}|\right\}} \right\}, \tag{3.20}$$

respectively. Now we are only left with identifying the correct form of the

residual R^{k+1} which is simply

$$R^{k+1} = \left\| \begin{bmatrix} \nabla^{\perp} V(\boldsymbol{x}_1^k) \\ \vdots \\ \nabla^{\perp} V(\boldsymbol{x}_N^k) \end{bmatrix} \right\|_{\infty} = \max_n \left\| \left\{ \nabla^{\perp} V(\boldsymbol{x}_n^k) \right\} \right\|_{\infty}. \tag{3.21}$$

To conclude, we present a summary of the method specifics for the dimer method introduced in Sec 2.9. To use the ode12 method, we consider the long vector

$$\mathbf{Z} := \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{bmatrix}, \tag{3.22}$$

in terms of the positions vector $\boldsymbol{x}$ and the orientation of the dimer $\boldsymbol{v}$, and the force term

$$\mathbf{F}_{\mathrm{D}} := \begin{bmatrix} \boldsymbol{f}_{\mathrm{x}} \\ \boldsymbol{f}_{\mathrm{v}} \end{bmatrix}, \tag{3.23}$$

in terms of the translational force $\boldsymbol{f}_{\mathrm{x}}$ and rotational force $\boldsymbol{f}_{\mathrm{v}}$, defined by equations (2.81) and (2.80), respectively,. The trial step is therefore given by

$$\mathbf{Z}^{k+1} = \mathbf{Z}^k + \alpha_k \mathbf{F}_{\mathrm{D}}^k, \tag{3.24}$$

the error meassure by

$$E^{k+1} = \frac{1}{2} \alpha_k (rtol)^{-1} \sup_j \left\{ \frac{|F_{\mathrm{D},j}^k - F_{\mathrm{D},j}^{k+1}|}{\max\left\{ \frac{atol}{rtol}, |X_j^k|, |X_j^{k+1}| \right\}} \right\}, \tag{3.25}$$

and the residual is given by

$$R^{k+1} = \|\boldsymbol{F}_{\mathrm{D}}\|_{\infty}. \tag{3.26}$$

3.2.3 Stagnation in the asymptotic regime

Applying the aforementioned ODE solver together with the string and NEB methods on a variety of systems, we observe a promising initial convergence behaviour. However, near the equilibrium, the method fails to select good step lengths for convergence. Instead, the method stagnates; a cycling behaviour is observed. To visualise, let us take a look at a vacancy migrating in a 249-atom

BCC W supercell, modelled with the EAM4 class of the EAM interatomic potential proposed by Marinica *et al.* [1] (see Sec. 2.3.1). The string method with free ends, coupled with the ode12 method on a path of 5 images is used. To initialise the path, the initial and final states $\boldsymbol{x}_1$ and $\boldsymbol{x}_5$ were relaxed to the local minima with an accuracy of 10^{-4}eV/Å and the path was assembled by linearly interpolating $\boldsymbol{x}_1$ and $\boldsymbol{x}_5$. Figure 3.1 demonstrates how ode12 effectively selects appropriate step lengths in the pre-asymptotic regime, but stagnates in the asymptotic regime.

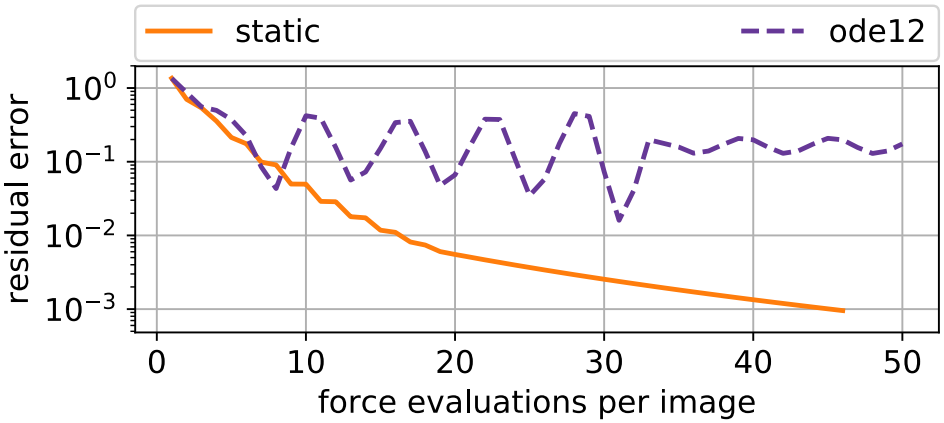


Figure 3.1: Convergence of the string method applied to vacancy migration in a 249-atom BCC W supercell modelled with the EAM4 potential [1]. Optimal static time stepping and time stepping with ode12 were used on a path consisting of 5 images.

Convergence of adaptive time stepping techniques has been studied excessively. As proved in Stuart [66], as $rtol \to 0$ a uniform convergence result holds, provided that the initial step is sufficiently small and that the solution is bounded away from the sets where $\|\mathbf{F}(\mathbf{X})\|$ and the leading error term (scaled by $\|\mathbf{F}(\mathbf{X})\|$),

$$\frac{1}{2}\|\mathbf{F}(\mathbf{X})\|^{-1}\mathbf{F}(\mathbf{X}) \cdot \nabla\mathbf{F}(\mathbf{X}), \tag{3.27}$$

vanish. It is important to notice here that our fictitious time coordinate is not bounded, instead we are intentionally approaching the zeros of $\|\mathbf{F}(\mathbf{X})\|$ and the convergence results of [67] for Runge-Kutta pairs $(p, p-1)$ of precisely p stages no longer hold. As discussed in [68], the reason the convergence results fail near equilibria, is because the step lengths increase to $\mathcal{O}(1)$ independently of the choice of $rtol$, and as a result, the higher order error term becomes of the

same order as the leading error term and cancellation may occur. In that case, the local error is underestimated and large time steps are falsely accepted, and the method fails to converge [66, 67, 68].

3.2.4 Minimising the residual

To overcome this issue, we modify ode12 described in Sec. 3.2.1, by adding a second step-length selection mechanism based on minimising the residual. In essence, the adaptive ODE step selection of Sec. 3.2.1 should be used in the pre-asymptotic regime while minimising the residual is a suitable mechanism in the asymptotic regime.

This leads to the following step-length selection algorithm, which we label *ode12r*:

We follow ode12 and calculate E^{k+1} from (3.13). We accept the proposed $\mathbf{X}^{k+1}$ if the scaled residual error satisfies either one of the two following conditions:

1) $R^{k+1} \leq R^k(1 - c_1\alpha_k)$,

2) $R^{k+1} \leq R^k c_2$ AND $E^{k+1} \leq rtol$,

for contraction and growth parameters c_1 and $c_2 \in \mathbb{R}$. Typical values for c_1 and c_2 are $c_1 = 0.01$ and $c_2 = 2.0$.

Whether the step is accepted or rejected, we now compute two step-length candidates using (1) the adaptive solver and (2) a simple line-search procedure. The adaptive solver step-length candidate given by the ode12 solver is $\alpha_{k+1,\text{ode12}} = \alpha_{k,\text{opt}}$ from (3.15). For the second candidate, let us consider the following argument: The driving force along the previous search direction can be approximated by its linear interpolant

$$\mathbf{F}(\mathbf{X}^k + \theta\alpha_k\mathbf{F}^k) = (1 - \theta)\mathbf{F}^k + \theta\mathbf{F}^{k+1} + \mathcal{O}(\alpha_k^2), \tag{3.28}$$

for some $\theta \in \mathbb{R}^+$. To ensure sufficient decrease of $\mathbf{F}$, we minimise

$$\|(1 - \theta)\mathbf{F}^k + \theta\mathbf{F}^{k+1}\|^2 \tag{3.29}$$

with respect to θ to obtain the line-search candidate

$$\alpha_{k+1,\text{ls}} = \theta\alpha_k, \tag{3.30}$$

where

$$\theta = \frac{\mathbf{F}^k \cdot (\mathbf{F}^k - \mathbf{F}^{k+1})}{\|\mathbf{F}^k - \mathbf{F}^{k+1}\|^2}. \tag{3.31}$$

If the current step $\mathbf{X}^{k+1}$ is accepted then the next step-length candidate is chosen to be

$$\alpha_{k+1} = \max\left(\tfrac{1}{4}\alpha_k, \min\left(4\alpha_k, \alpha_{k+1,\text{ls}}, \alpha_{k+1,\text{ode12}}\right)\right),$$

so that it is not too small or too large compared to the last step α_k. If the step $\mathbf{X}^{k+1}$ is rejected, then the new step-length candidate starting from $\mathbf{X}^k$ is

$$\alpha_k = \max\left(\tfrac{1}{10}\alpha_k, \min\left(\tfrac{1}{4}\alpha_k, \alpha_{k+1,\text{ls}}, \alpha_{k+1,\text{ode12}}\right)\right),$$

so that it is not too dissimilar to the rejected step.

In Fig. 3.2, the convergence speed of the modified *ode12r* agrees with the results of ode12 in the pre-asymptotic regime but successfully converges upon reaching the asymptotic regime. Highlighting our earlier comment; near the equilibrium, the system is already on a descent direction and we only need to make sure that we approach the solution sufficiently fast and we do so by contracting the residual, whereas if the system is starting far from the stable solution, by following the ode adaptive step we are guaranteed to follow a stable, descent direction.

3.3 Results

In this section we look at the convergence results of the *ode12r* ODE schemes and the linesearch method. The measure for convergence is the magnitude of the residual error R^{k+1} as defined in Eqs. (3.16), (3.21) and (3.26). To demonstrate its effectiveness, we first look at the Müller-Brown potential (see Sec. 2.2.1) and we carry on to study a set of systems in interatomic potentials. The description of the systems can be found in Sec. 2.3.

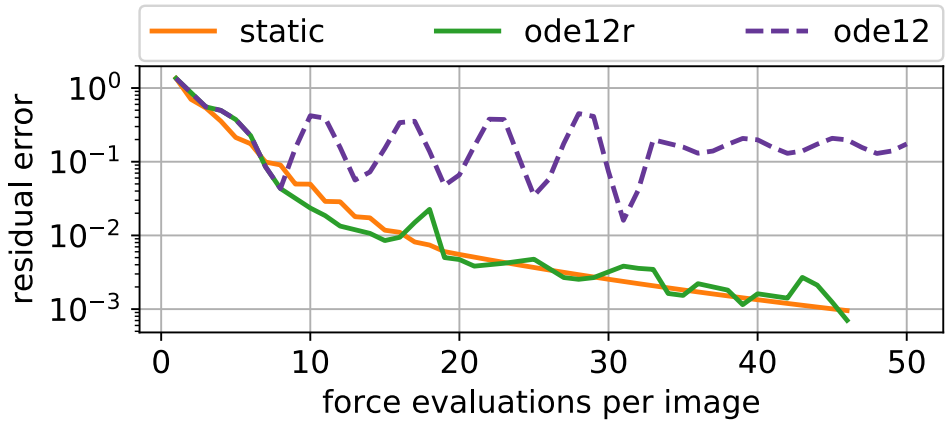


Figure 3.2: Convergence rate of the string method applied to vacancy migration in a 249-atom BCC W supercell modelled with the EAM4 potential [1]. Optimal static time stepping, time stepping with ode12 and time stepping with *ode12r* were used with a path consisting of 5 images.

Since the preliminary target of these methods is adaptivity, the main focus of the following section is the study of robustness of the methods in terms of parameter selection.

For each example, we explore the methodologies in conjunction with the general optimisation problem, where we compare the *ode12r* method against steepest descent, the LBFGS and conjugate gradient methods. For the LBFGS and conjugate gradient methods the implementation of the Optim.jl package [69] was used. We then move to the string and NEB methods and conclude with the dimer method.

3.3.1 Müller-Brown potential

We first look at the Müller-Brown potential, a two-dimensional potential surface, which we introduced in Sec. 2.2.1. The surface has three local minima which we labeled $\boldsymbol{x}_\mathrm{A}$, $\boldsymbol{x}_\mathrm{B}$ and $\boldsymbol{x}_\mathrm{C}$ on Fig. 2.4 and two saddle points labeled $\boldsymbol{x}_{S_1}$ and $\boldsymbol{x}_{S_2}$.

We start from a state $\boldsymbol{x}_0$ in the vicinity of $\boldsymbol{x}_\mathrm{A}$ and we use *ode12r* to minimise the energy to $\boldsymbol{x}_\mathrm{A}$. Figure 3.3 shows the initial state $\boldsymbol{x}_0$ marked with the darkest shade of red and the local minimum $\boldsymbol{x}_\mathrm{A}$ marked in light yellow. The sequence of steps taken is marked with gradually varying colour from red to yellow.

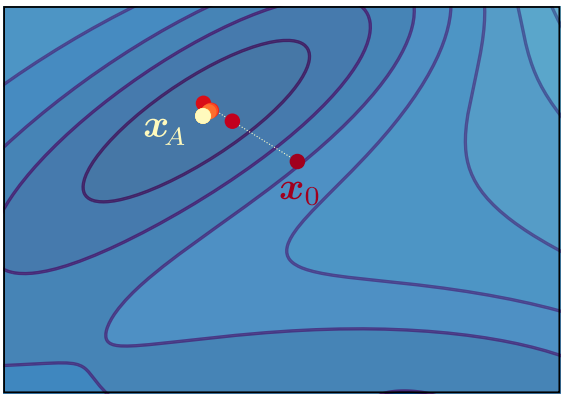


Figure 3.3: Contours of the Müller-Brown potential. Starting from a state $\boldsymbol{x}_0$, we relax the configuration to local minimum $\boldsymbol{x}_\mathrm{A}$ with *ode12r*. The sequence of steps taken by *ode12r* is marked with gradually varying colour from red to yellow.

A comparison of the results of the *ode12r* and ode12 algorithms is shown in Fig. 3.4(a). Notice how using a very fine relative tolerance allows ode12 to eventually converge but this requires considerably more computational time. The additional residual minimisation step of the *ode12r* algorithm takes advantage of the fast convergence of the adaptive step for residual errors of the order of $\sim 10^{-1}$ and permits further reduction of the residual error for residual errors beyond 10^{-1}. A comparison against the widely used steepest descent, LBFGS and conjugate gradient optimisation tools is presented in Fig. 3.4(b), where the *ode12r* method exhibits a 2-fold speedup compared to the most optimal, in this case, conjugate gradient method.

To demonstrate the robustness of the *ode12r* method, we run the *ode12r* method, from the same initial state, varying the values of *rtol* and *atol* in $\{a \times 10^b : a = 1, \ldots, 9, b = 0, -1 \ldots, -4\}$. Figure 3.5 shows the number of force calls needed to converge to an accuracy of 10^{-4} for the different values of *rtol* and *atol*.

The string and NEB methods were used to find the MEP from state $\boldsymbol{x}_\mathrm{A}$ to state $\boldsymbol{x}_\mathrm{B}$; refer to Fig 2.10 for a visualisation of this. Remember from Sec. 2.4, that the MEP passes through $\boldsymbol{x}_\mathrm{C}$ and both saddle points $\boldsymbol{x}_{S_1}$ and $\boldsymbol{x}_{S_2}$. Although this violates the assumptions posed in Sec. 2.8 on the uniqueness of the saddle, the string and NEB methods were able to tackle this problem with no issue because the two saddles are sufficiently far apart. One can always

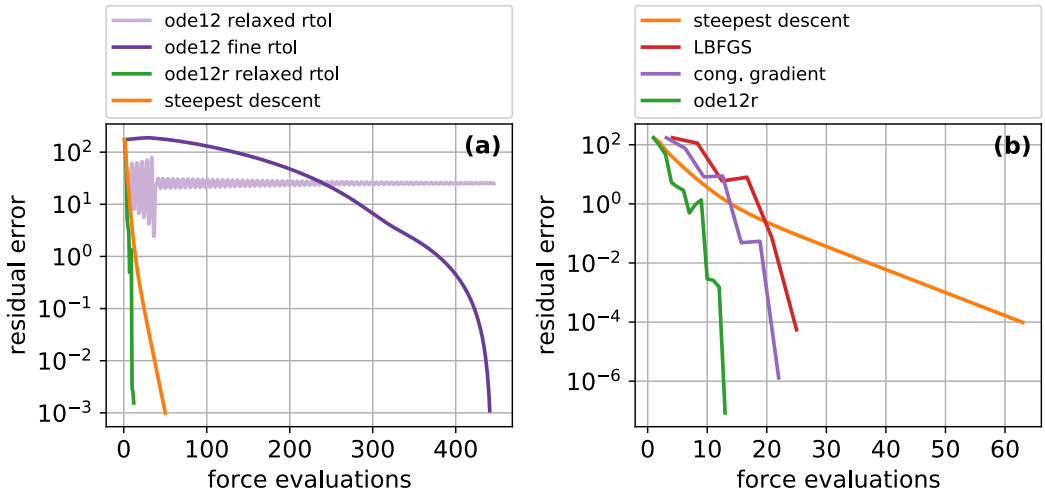


Figure 3.4: Convergence (in terms of the number of force evaluations) for the optimisation problem for a system in the Müller-Brown potential. On the left pane (a), the *ode12r* method is compared against ode12 both using the same tolerances and ode12 with a finer tolerance. On the right hand pane (b), time stepping with *ode12r* is compared against optimally tuned steepest descent method, LBFGS and conjugate gradient methods.

consider the transition as the sum of two transitions from $\boldsymbol{x}_{\mathrm{A}}$ to $\boldsymbol{x}_{\mathrm{C}}$ and from $\boldsymbol{x}_{\mathrm{C}}$ to $\boldsymbol{x}_{\mathrm{B}}$. Table 3.1 focuses on the computational cost of the *ode12r* and static string and NEB methods for a low accuracy of $\sim 10^{-1}$ and a high accuracy of $\sim 10^{-3}$. For the static string and NEB the step length was carefully chosen to achieve the fastest convergence to the corresponding low and high accuracies. We refer to this choice as the optimally chosen static step. For the *ode12r* adaptive step selection, the relative tolerance was set to 10^{-1} and the absolute tolerance was chosen to match this value. The *ode12r* step selection was found to be at least as good as the optimally chosen static step but beyond this it exhibited a 2-fold speedup for the NEB method for higher accuracies. Finally, we test the dimer method on finding saddle $\boldsymbol{x}_{\mathrm{S}_1}$ starting from a point along the line connecting $\boldsymbol{x}_{\mathrm{A}}$ and $\boldsymbol{x}_{\mathrm{S}_1}$. The convergence results for finding the saddle using the dimer method, are summarised in Table 3.2, which focuses on the computational cost of the *ode12r* and simple dimer method for a low accuracy of $\sim 10^{-1}$ and a high accuracy of $\sim 10^{-4}$. The advantage from using adaptivity to select the step length is clear. A 2-fold speedup was observed for the low accuracy, whereas a speedup of almost a factor of 4 was observed for the higher

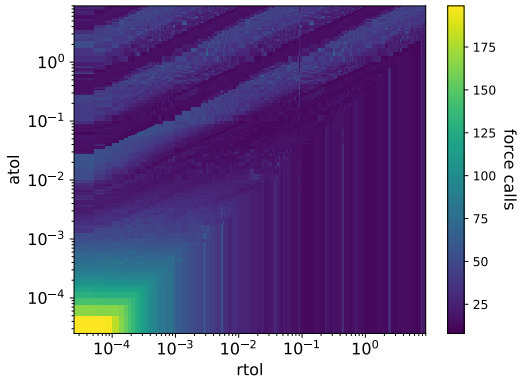


Figure 3.5: Variation of computational cost of the *ode12r* parameters, for the optimisation problem applied to the Müller-Brown potential. The system was initialised from the same state $\boldsymbol{x}^0$ near the equilibrium for each run.

Müller-Brown potential: MEP finding methods				
Step selection	static		*ode12r* solver	
Tol	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	45	56	43	54
NEB	41	75	33	44

Table 3.1: Force evaluations per image needed for the string and NEB methods to converge to the MEP connecting the two local minima $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ of the Müller-Brown potential. The MEP was discretised by 15 images.

accuracy.

3.3.2 Vacancy

We explore vacancies in three different systems. These were chosen to make critical comparisons between the systems and give an insight on the capabilities and limitations of the ode12 method.

Vacancy in a 2D Lennard-Jones cell

Consider a vacancy in a 60-atom triangular lattice modelled with the Lennard-Jones potential with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$, as described in Sec. 2.3.1. The vacancy is located at the centre of the cell initially and periodic boundary conditions are imposed in the x and y directions.

From a state close to the local minimum, we first consider the min-

Müller-Brown potential: walker type method				
Step selection	simple dimer		*ode12r* solver	
Tol	10^{-1}	10^{-4}	10^{-1}	10^{-4}
Dimer	53	101	22	28

Table 3.2: Force evaluations needed for the dimer method to converge to saddle $\boldsymbol{x}_{S_1}$ of the Müller-Brown potential.

imisation problem to the local minimum. The results are shown in Fig. 3.6. Figure 3.6(a) highlights how *ode12r* uses the ode12 step length predictions in the pre-asymptotic regime, before line search step length selection takes over in the asymptotic regime. Notice that choosing a fine *rtol* tolerance for ode12 will result to the method converging after a long time. We also look

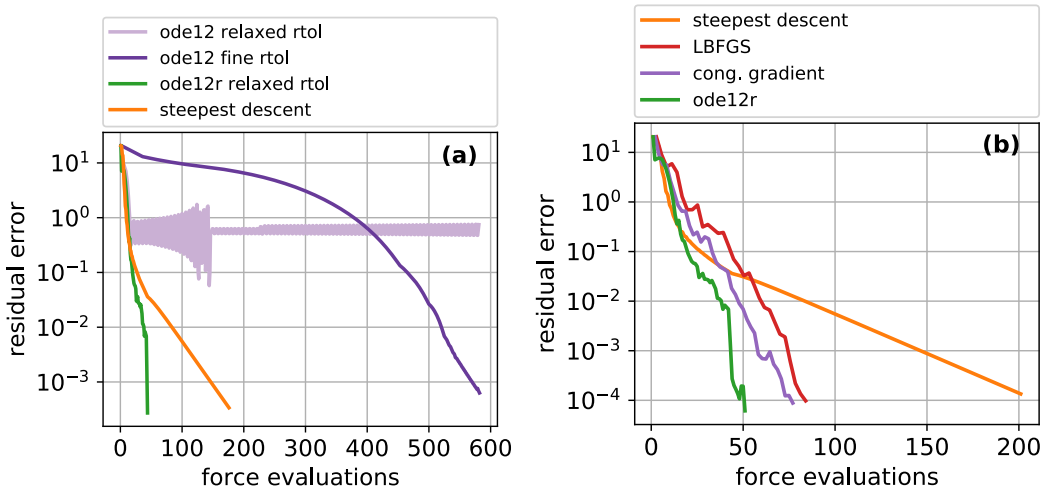


Figure 3.6: Convergence of optimising the configuration in a 60-atom 2D cell modelled with a Lennard-Jones potential and containing a vacancy. The left panel (a) shows a comparison of *ode12r* to ode12 and the right panel (b) shows comparison against steepest descent, LBFGS and conjugate gradient.

at parameter fitting for the *atol* and *rtol* parameters. We run the *ode12r* method with the same initialisation, varying the values of *rtol* and *atol* in $\{a \times 10^b : a = 1, \ldots, 9, b = 0, -1 \ldots, -4\}$. Figure 3.7 shows the number of force calls needed to converge to an accuracy of 10^{-4} for the optimisation algorithm, as a function of *rtol* and *atol*.

We now look at the transition of the vacancy migrating by one lattice spacing in the y direction. Considering the string and NEB methods to find

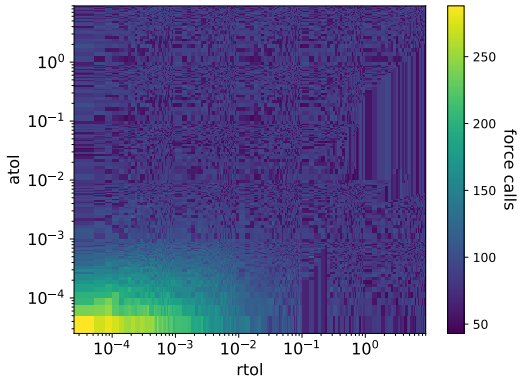


Figure 3.7: Variation of computational cost with the *ode12r* parameter, for the optimisation algorithm applied to a 60-atom 2D cell modelled with a Lennard-Jones potential containing a vacancy. The system was initialised at a state near the local minimum.

the MEP, we tabulate the number of force calls per image required for convergence for either of the static and adaptive *ode12r* step length selection schemes. Table 3.3 shows these results. Neither method converges beyond the 10^{-3} tolerance within a reasonable number of iterations. The latter came as a surprise to us, as on the contrary to the real vacancy migration systems that we study next, this artificial set up appears to exhibits more severe ill-conditioning. Figure 3.8, shows the number of force calls per image needed for the string method with *ode12r* to converge to an accuracy of just 8×10^{-2}eV/Å, as a function of *atol* and *rtol*. On this colour map, regions in white show that the method did not converge within a reasonable number of force calls. We will revisit this again in Chapter 4.

2D Vacancy: MEP finding methods

Step selection	static		*ode12r* solver	
Tol	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	197	*	52	*
NEB	200	*	53	*

Table 3.3: Number of force evaluations per image required by the string and NEB methods to converge the vacancy migration MEP of a 60-atom 2D cell modelled with a Lennard-Jones potential, with either the static or *ode12r* step length selection methods. In the cases marked *, the algorithm did not converge within a reasonable number of iterations.

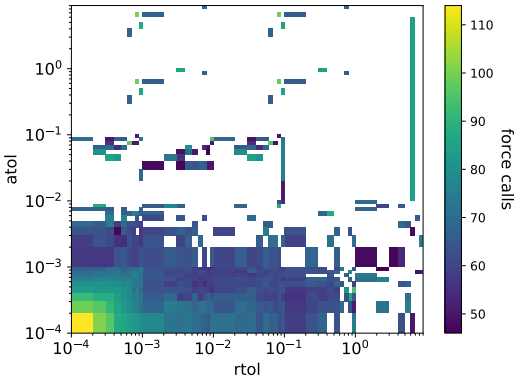


Figure 3.8: Variation of computational cost of the *ode12r* parameters, for the string method applied to vacancy migration in a 60-atom 2D triangular lattice modelled with a Lennard-Jones potential. The string method was converged to an accuracy of 8×10^{-2}eV/Å. White indicates cases where the string method did not converge within a reasonable number of force computations.

Next we turn our attention to three dimensional systems containing a vacancy. We consider a 107-atom Cu FCC supercell in a fixed cell, periodic boundary conditions and interactions modelled with a Morse potential with parameters $A = 4.0$, $\epsilon = 1.0$ and nearest neighbour distance $r_0 = 2.55$ Å, as described in Sec. 2.3.1.

For the optimisation problem, we considered two initial conditions. For the first, we initialise our system far from a local minimum of the energy and for the second case we begin optimisation from a state close to the local minimum. We conduct the same parameter fitting analysis for the *ode12r* method that we carried on earlier for the vacancy in Lennard-Jones potential. Figure 3.9 shows the number of force calls needed to converge to an accuracy of 10^{-4}eV/Å for the different values of *rtol* and *atol*. It is evident that small changes in the values of the tolerance parameters will not cause dramatic changes for the computational cost. The convergence results of the optimisation for ode12 and *ode12r* are presented in Fig. 3.10 along with the convergence rates of the static steepest descent, the LBFGS method and the conjugate gradient method. The effectiveness of the line search in *ode12r* is illustrated on the left hand side panels (a) and (c): for the simple optimisation problem, both initial conditions are in the asymptotic regime and therefore the line search takes effect immediately. Comparing *ode12r* against other methods

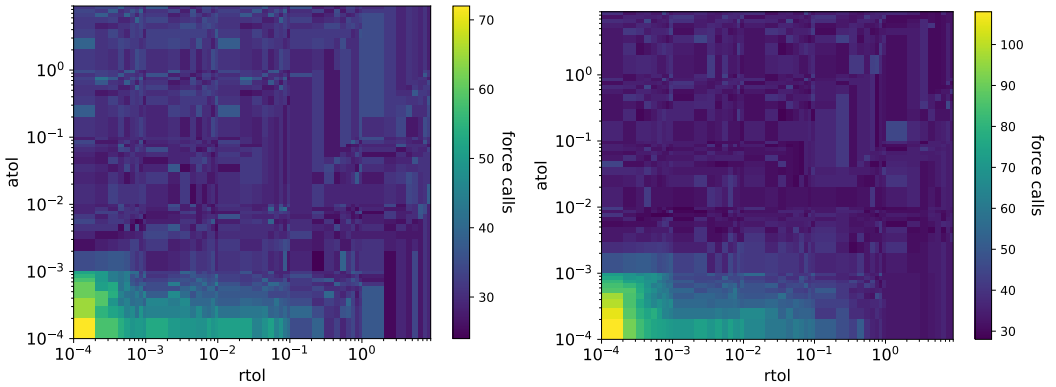


Figure 3.9: Variation of computational cost with the *ode12r* parameter, for the optimisation problem applied to vacancy migration in a 127-atom fcc Cu supercell modeled with the Morse potential [1]. The system was initialised with a state far from the local minimum on the right pane and near the minimum on the left panes.

in Figs 3.10(b),(d), we see a speed up of factor at least 2 in comparison to the simple steepest descent method but also a substantial speed up in comparison to the conjugate gradient method. Notice here that there is a greater speed up for the system initialised furthest from the minimum than for the system initialised close to the equilibrium. Next we couple *ode12r* with the string

Vacancy in Cu supercell: MEP finding methods

Step selection	static		*ode12r* solver	
Tol / eV/Å	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	8	74	8	41
NEB	8	57	8	27

Table 3.4: Force evaluations per image needed for the string and NEB methods for the migration of a vacancy in a 107-atom Cu FCC supercell modelled by a Morse potential. The MEP was discretised with 5 images.

and NEB methods. A path of 5 images was used, and the end points were left free to move. To initialise the path, the reactant and product states were relaxed to an accuracy of 10^{-4}eV/Å and the initial path was constructed from their linear interpolation. Table 3.4 shows the number of force evaluations per image needed for convergence to two preset tolerance limits. Figure 3.11 illustrates once again how the line search step takes over the adaptive ode step in order to aid convergence. It is important to note here that in fact for the

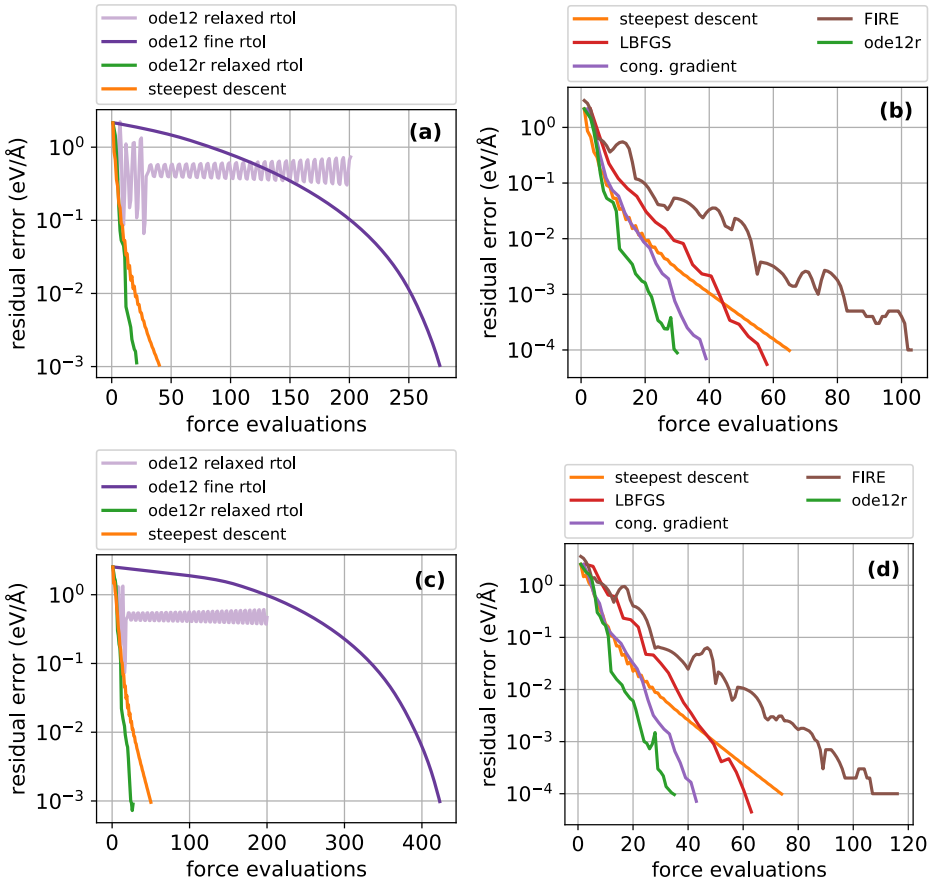


Figure 3.10: Convergence for the optimisation problem applied to vacancy migration in a 127-atom FCC Cu supercell modelled with the Morse potential [1]. The methods were initialised with a state near the local minimum on panes (a) and (b) and far from the minimum on the lower panes (c) and (d). On the left panes, optimally tuned steepest descent method, time stepping with ode12 and time stepping with *ode12r* were used. On the right panes, the comparison includes the LBFGS, conjugate gradient and FIRE methods, for the latter, the implementation from the ASE [3] library is used.

string method the adaptive ode step is never activated. This example demonstrates how the *ode12r* solver can aid the performance of the string and NEB methods if a static step is not suitable.

We also explore the effectiveness of *ode12r* with the dimer method. We start from a configuration found close to the midpoint between the reactant state and the saddle point configuration of the system. We summarise our findings in Table 3.5. Convergence to low accuracies is the same for the two

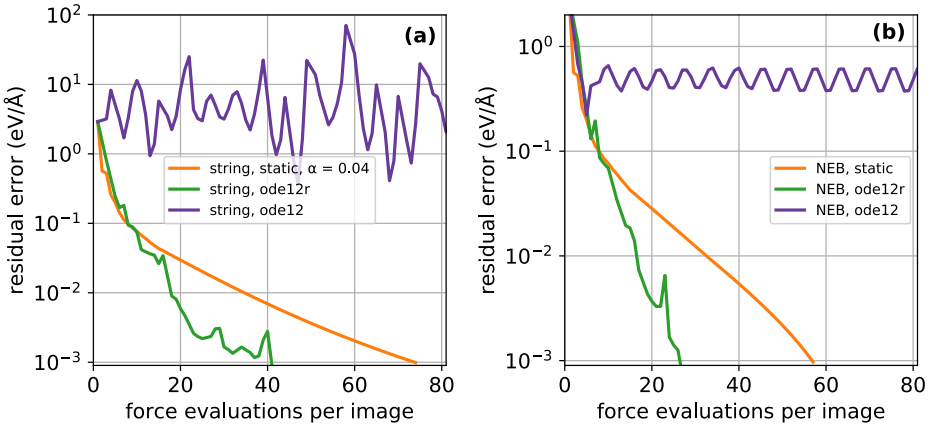


Figure 3.11: Convergence of the (a) string and (b) NEB methods for vacancy migration in a 107-atom Cu FCC supercell modelled by a Morse potential [1]. The MEP was discretised with 5 images.

Vacancy in Cu supercell: walker type method				
Step selection	static		*ode12r* solver	
Tol / eV/Å	10^{-1}	10^{-4}	10^{-1}	10^{-4}
Dimer	8	83	8	34

Table 3.5: Force evaluations needed for convergence of the dimer method to the saddle point of a vacancy migrating in a 107-atom Cu FCC supercell modelled by a Morse potential.

step selection schemes, but the *ode12r* achieved a speedup of the order of at least a factor of 2 on higher accuracies.

3.3.3 Conclusions

We have proposed a new adaptive scheme for step length selection suitable for saddle point finding methods such as the string and NEB methods and the dimer method. The proposed technique is motivated by the observation that saddle point finding methods are expressed as steady state solutions to first order ODEs. This allowed the use of the adaptive ODE solver ode12 which uses a first and a second order approximation of a solution to predict a suitable step length for the next iteration. The ODE solver was constructed following Hairer *et al.* [65], and it requires the user to supply an absolute and a relative tolerance *atol* and *rtol*. We found that the latter are in fact easy to choose.

However, as proved by Stuart [66], the convergence of such an adaptive ODE solver is only guaranteed in regions away from the equilibria. But in our case we pursue converging to the equilibrium where the forcing term vanishes. This is why we modify the step selecting scheme, so that line search is performed at each iteration, which ensures sufficient decrease of the magnitude of the force term.

The developed scheme could in fact be applied to general optimisation, giving a robust adaptive step selection scheme for steepest descent.

In our results, we observed that fitting the *rtol* and *atol* parameter was simple, as it was observed that $rtol = 0.1$ was sufficient in most cases for convergence but other values $rtol = 1$ and $rtol = 0.01$ were occasionally more appropriate. The value of *atol* was chosen so that $atol/rtol = 1$ in all cases except for the string method of the 2D vacancy, where $atol/rtol = 0.01$ had to be used instead. Diagrams showing the convergence of the *ode12r* scheme for a range of *rtol* and *atol* values were explored, indicating that the method is very robust in general, with the exception of the string/NEB methods applied to vacancy migration in Lennard-Jones potential where the method failed to converge to a high accuracy.

The results of this chapter were published as part of Makri *et al.* [70]. An open source prototype implementation of our technique is available at `https://github.com/cortner/SaddleSearch.jl`.

Chapter 4

Preconditioning

4.1 Overview

In this chapter, we revisit ill-conditioned potentials, which were first discussed in Chapter 2 and we extend our framework to discuss the implications of an ill-conditioned potential to the NEB and string methods. We review the *preconditioning method* for numerical optimisation which will motivate our preconditioned string and NEB scheme; an effective scheme which reduces the computational cost attributed to ill-conditioning. We note here that an effective preconditioning scheme for the dimer method has already been developed by Gould *et al.* [16]. We conclude by revisiting the adaptive step selection method *ode12r* of Chapter 3 to further improve the robustness of the proposed method. We demonstrate the effectiveness of this combination on a range of materials modelling examples.

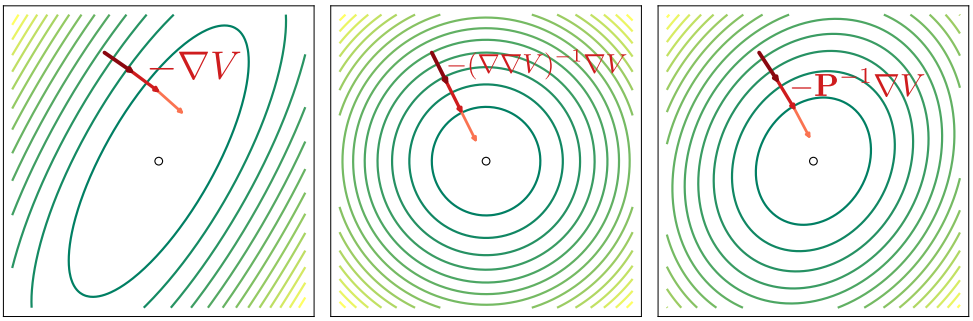
Recall from (2.59) that the condition number of a twice differentiable potential V is given by $\kappa(V) = \max_m |\lambda_m| / \min_m |\lambda_m|$ and that this quantity is greater than or equal to 1. Recall further that a large condition number, gives slow convergence for the steepest descent method. To visualise, the spread of the eigenspectrum of the Hessian $\nabla\nabla V$ is large, thus, in the neighbourhood of a minimum, the contours of V are skew-elliptical, and hence taking steps in the steepest descent direction might result in missing the minimum altogether. This is illustrated best in the 2D diagram of Fig. 4.1a. But we have seen in Sec. 2.6 that this is not the case for the Newton direction $-(\nabla\nabla V(\boldsymbol{x}))^{-1}\nabla V(\boldsymbol{x})$ of expression (2.45).

Preconditioning is a method from linear algebra, which is motivated from the above observation. In particular, in steepest descent optimisation, preconditioning has two related but distinct interpretations [15]:

1. as an approximation of the Hessian; $\mathbf{P} \approx \nabla\nabla V$, is used in analogy to Newton's scheme to *precondition* the steepest descent direction according to
$$-\nabla V(\boldsymbol{x}) \mapsto -\mathbf{P}^{-1}\nabla V(\boldsymbol{x}), \tag{4.1}$$

2. or as a coordinate transformation in the state space
$$\boldsymbol{x} \mapsto \mathbf{P}^{1/2}\boldsymbol{x}, \tag{4.2}$$
that captures information of the local curvature of the potential landscape, mapping hyperellipsoids to balls.

In the second interpretation, one understands the transformation as an attempt to reduce the condition number of V in the transformed space. Figure 4.1c illustrates the geometric considerations of preconditioning.



(a) steepest descent (b) Newton's method (c) preconditioned steepest descent

Figure 4.1: The diagrams show the contours of an ill-conditioned function V. In diagram (a) the contours near the minimum are skew-elliptical. Taking steepest descent steps, marked in red in the schematic, results in diverging from the local minimum. In diagram (b) Newton's method is used instead. This is equivalent to locally mapping the elliptic contours to circles, alowing fast convergence to the minimum. Diagram (c) illustrates how the use of a preconditioner $\mathbf{P} \approx \nabla\nabla V$ aids convergence to the local minimum.

A suitable preconditioner, $\mathbf{P} \in \mathbb{R}^{M\times M}$, should be cheap to compute,

as opposed to the construction of the true Hessian matrix $\nabla\nabla V$ and positive definite everywhere in V so that a descent direction is mapped to a descent direction. It has been shown for example in [2, 71, 72] how to construct and invert effective preconditioners for the potential energy landscape of materials and molecules at a cost comparable to the evaluation of an interatomic potential and much lower than the cost of evaluating a DFT model. When used correctly, preconditioning leads to a substantial reduction in the number of force calls and thus, provided that the computational cost of a preconditioner evaluation is comparable in computation time to a model computation, it is expected to significantly improve computing times [73, 71].

4.2 Ill-conditioning and MEP finding

Since the NEB and string methods evolve a path along a projected steepest descent direction, inheriting from the steepest descent method, they inevitably exhibit slow convergence rates when they are subjected to ill-conditioned energy landscapes V. In the following, we describe a simple yet effective way to precondition the standard NEB and string methods to obtain efficient and robust algorithms for computing MEPs in ill-conditioned geometries. Care needs to be taken as a preconditioner encapsulating the curvature of the potential will in general be variable in phase-space $\boldsymbol{x}$. As a direct consequence, each image along a discretised path of the NEB and string method requires a different preconditioner matrix. The difficulty arrises when projecting the potential gradient along a tangent and in finding an appropriate norm to describe distances along the transformed space.

The same preconditioners used in geometry optimisation of interatomic potentials [71, 2] are expected to be valid for the purposes of preconditioning each image separately.

4.2.1 Preconditioned String Method

We first present our construction of the preconditioned string method which has a simpler updating step.

Let us first consider the simple case where $\mathbf{P}$ is constant in $\boldsymbol{x}$. Starting

from the coordinate transformation

$$\boldsymbol{x} \mapsto \mathbf{P}^{1/2}\boldsymbol{x} := \tilde{\boldsymbol{x}}, \tag{4.3}$$

with corresponding

$$\tilde{V}(\tilde{\boldsymbol{x}}) = V(\mathbf{P}^{-1/2}\tilde{\boldsymbol{x}}). \tag{4.4}$$

This allows us to express the updating step of the string method in the transformed space, as

$$\tilde{\boldsymbol{x}}_n^{k+1} = \tilde{\boldsymbol{x}}_n^k - \alpha^k \nabla^{\perp} \tilde{V}(\tilde{\boldsymbol{x}}_n^k). \tag{4.5}$$

For convenience we rewrite equation (4.5) as

$$\begin{aligned} \tilde{\boldsymbol{x}}_n^{k+1} &= \tilde{\boldsymbol{x}}_n^k - \alpha^k \left(\mathbf{I} - \tilde{\boldsymbol{t}}_n^k \otimes \tilde{\boldsymbol{t}}_n^k\right) \nabla_{\tilde{x}} \tilde{V}(\tilde{\boldsymbol{x}}_n^k), \\ \tilde{\boldsymbol{t}}_n^k &= \frac{(\tilde{\boldsymbol{x}}_n^k)'}{\|(\tilde{\boldsymbol{x}}_n^k)'\|}. \end{aligned} \tag{4.6}$$

It is trivial to deduce that, componentwise,

$$\frac{\partial \tilde{x}_i}{\partial x_j} = P_{ij}^{1/2}, \text{ and } \frac{\partial x_i}{\partial \tilde{x}_j} = P_{ij}^{-1/2}. \tag{4.7}$$

Then, reversing the coordinate transformation we obtain an equivalent formulation in the original coordinates with updating step

$$\begin{aligned} \boldsymbol{x}_n^{k+1} &= \boldsymbol{x}_n^k - \alpha^k \left(\mathbf{P}^{-1} - \boldsymbol{t}_{\mathrm{P},n}^k \otimes \boldsymbol{t}_{\mathrm{P},n}^k\right) \nabla_x V(\boldsymbol{x}_n^k), \\ \boldsymbol{t}_{\mathrm{P},n}^k &= \frac{(\boldsymbol{x}_n^k)'}{\|(\boldsymbol{x}_n^k)'\|_{\mathbf{P}}}. \end{aligned} \tag{4.8}$$

where care needs to be taken to normalise the tangents $\boldsymbol{x}'$ with respect to the $\mathbf{P}$-norm, $\|\mathbf{y}\|_{\mathbf{P}} = (\mathbf{y} \cdot \mathbf{P}\mathbf{y})^{1/2}$, instead of the usual ℓ^2-norm, $\|\mathbf{y}\| = (\mathbf{y} \cdot \mathbf{y})^{1/2}$.

In this simple case, expressing the reparameterisation step in terms of coordinates in the configuration space is trivial, as it suffices to replace the usual ℓ^2-norm with the $\mathbf{P}$-norm, due to linearity of the $\frac{\mathrm{d}}{\mathrm{d}s}$ operator. The details of the preconditioned reparameterisation algorithm are omitted for now, as we will be revisiting the algorithm in Sec. 4.2.2.

The systems of interest, however, are described by preconditioners that are not constant in the configuration space [71]. This leads to a Riemannian

metric framework. In particular, expressing the general Eq. (4.6) in the original coordinates in analogy to Eq. (4.8), now takes the form

$$
\begin{aligned}
\boldsymbol{x}_n^{k+1} &= \boldsymbol{x}_n^k - \alpha^k \mathbf{P}^{-\frac{1}{2}}(\boldsymbol{x}_n^k)\Big(\mathbf{I} - \tilde{\boldsymbol{t}}_n^k \otimes \tilde{\boldsymbol{t}}_n^k\Big)\nabla_{\boldsymbol{x}} V|_{\boldsymbol{x}_n^k} \cdot \frac{\partial \boldsymbol{x}_n^k}{\partial \tilde{\boldsymbol{x}}_n^k}, \qquad (4.9)\\
\tilde{\boldsymbol{t}}_n^k &= \frac{\frac{\partial \tilde{\boldsymbol{x}}_n^k}{\partial \boldsymbol{x}_n^k}(\boldsymbol{x}_n^k)'}{\left\| \frac{\partial \tilde{\boldsymbol{x}}_n^k}{\partial \boldsymbol{x}_n^k}(\boldsymbol{x}_n^k)' \right\|},\\
\frac{\partial \tilde{\boldsymbol{x}}_n^k}{\partial \boldsymbol{x}_n^k} &= \nabla_{\boldsymbol{x}}(\mathbf{P}^{1/2}\boldsymbol{x})|_{\boldsymbol{x}_n^k},\\
\frac{\partial \boldsymbol{x}_n^k}{\partial \tilde{\boldsymbol{x}}_n^k} &= \left(\nabla_{\boldsymbol{x}}(\mathbf{P}^{1/2}\boldsymbol{x})|_{\boldsymbol{x}_n^k}\right)^{-1},
\end{aligned}
$$

which involves the evaluation of the full $\nabla_{\boldsymbol{x}}(\mathbf{P}^{\frac{1}{2}}\boldsymbol{x})|_{\boldsymbol{x}_n^k}$ matrix, which in general is algebraically complex and computationally expensive[1].

We circumvent these issues entirely by dropping these terms. We justify this simplification on the ground that $\mathbf{P}$ changes 'slowly' along a path, in the sense that assuming the neighbouring images of $\boldsymbol{x}_n$ are sufficiently far, there is a small neighbourhood around $\boldsymbol{x}_n$ where $\mathbf{P}$ is approximately constant.

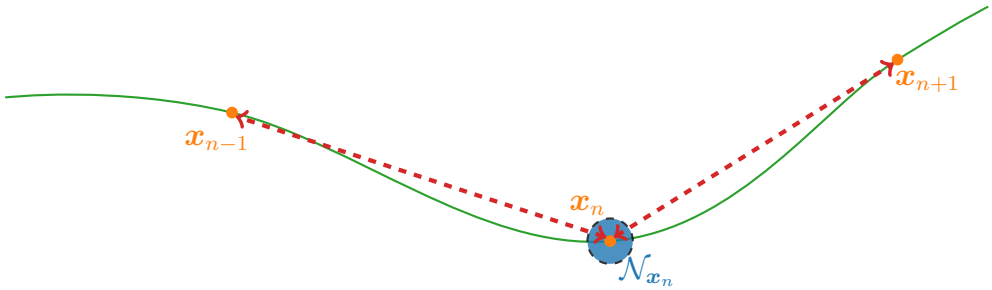


Figure 4.2: Illustration of the neighbourhood $\mathcal{N}_{\boldsymbol{x}_n}$ of $\boldsymbol{x}_n$. $\mathbf{P}$ is assumed to be approximately constant and equal to $\mathbf{P}(\boldsymbol{x}_n)$ in $\mathcal{N}_{\boldsymbol{x}_n}$. The neighbouring images $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_{n+1}$ are sufficiently far from $\boldsymbol{x}_n$.

Preliminary tests showed that this simplification does not lead to any loss of performance (see Fig. 4.3 later on). Thus, we obtain the preconditioned string method

$$
\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k - \alpha^k \nabla^{\perp} V_{\mathrm{P}}(\boldsymbol{x}_n^k), \qquad (4.10)
$$

[1]To evaluate $\nabla_{\boldsymbol{x}}(\mathbf{P}^{\frac{1}{2}}\boldsymbol{x})|_{\boldsymbol{x}_n^k} = \mathbf{P}^{\frac{1}{2}} + \nabla_{\boldsymbol{x}}\mathbf{P}^{\frac{1}{2}}|_{\boldsymbol{x}_n^k}\boldsymbol{x}_n^k$, we need to compute first $\mathbf{P}^{\frac{1}{2}}$ and $\nabla_{\boldsymbol{x}}\mathbf{P}^{\frac{1}{2}}|_{\boldsymbol{x}_n^k}$ which is difficult and computationally expensive.

where we define the quantities

$$\nabla^{\perp} V_{\mathrm{P}}(\boldsymbol{x}_n^k) = \left([\mathbf{P}_n^k]^{-1} - \boldsymbol{t}_{\mathrm{P},n}^k \otimes \boldsymbol{t}_{\mathrm{P},n}^k\right) \nabla_x V(\boldsymbol{x}_n^k), \tag{4.11}$$

$$\boldsymbol{t}_{\mathrm{P},n}^k = \frac{(\boldsymbol{x}_n^k)'}{\|(\boldsymbol{x}_n^k)'\|_{\mathbf{P}_n^k}}, \tag{4.12}$$

in terms of the $\mathbf{P}_n^k = \mathbf{P}(\boldsymbol{x}_n^k)$.

We illustrate the above by including a test carried out on the string method, in Fig. 4.3, for a vacancy migration confined on a 53-atom W BCC supercell, modelled by the EAM4 potential [1] with periodic boundary conditions (as discussed in Sec. 2.3.1). For preconditioning, the force field preconditioner (FF) [2] is uded, which is introduced later on in Sec. 4.4.2. The performance (in terms of force evaluations) of the full and simplified preconditioned schemes are comparable. The schemes are indeed comparable in performance. We are using a one-step method, and if the energy was to change dramatically so that the neglected terms were important, one should have taken a shorter step; thus the results of Fig. 4.3 show what we expect to see. The important thing to remember is that we do not require the preconditioning to describe the system perfectly, but rather we are interested in the performance it offers. The proposed simplification offers sufficient convergence without any additional computational overhead from constructing the preconditioner itself.

4.2.2 Reparameterising in the preconditioned string

We are left to specify how to reparameterise the path. The path reparameterisation described in Eq. 12 in [11] assumes that the ℓ^2-metric is used to measure distance. Recall, in Ch. 2 we saw that in the continuous limit, we are free to use any parameterisation for the path. In our setting, the premise is that $\|\cdot\|_{\mathbf{P}}$ is a more natural notion of distance than the standard ℓ^2-norm $\|\cdot\|$. We define the following notion of distance along the path:

$$\mathrm{d}_{\mathbf{P}}(\boldsymbol{x}, \boldsymbol{y}) := \left((\boldsymbol{x} - \mathbf{y}) \cdot \left(\frac{\mathbf{P}(\boldsymbol{x}) + \mathbf{P}(\mathbf{y})}{2} \right) (\boldsymbol{x} - \mathbf{y}) \right)^{1/2}. \tag{4.13}$$

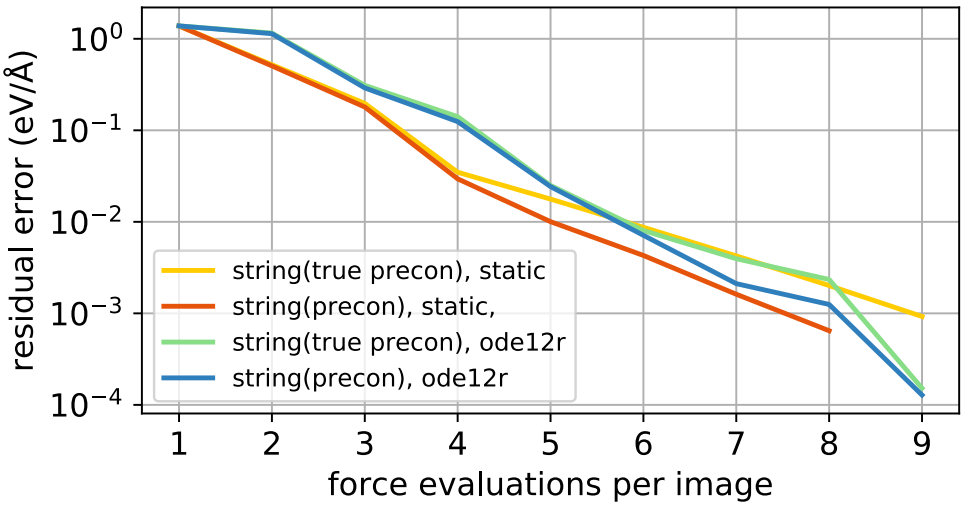


Figure 4.3: Convergence of the full analytic expression for the preconditioned string method (labeled in the figure as *"true precon"*) and the simplified preconditioned string (labeled as *"precon"*) proposed in equations (4.10) and (4.11). The system is a 53-atom W bcc supercell modelled by the EAM4 potential [1]. The method is applied to find the MEP of the migration of the vacancy. The path was discretised by 5 images and the preconditioner was constructed from the force field [2].

We note that $\mathrm{d}_{\mathbf{P}}$ is *not* a metric in the technical sense, as it does not satisfy the triangle inequality. However, it is an approximation (discretisation) of the geodesic distance on the Riemannian manifold induced by the preconditioner $\mathbf{P}$. From this, it is reasonable to expect that $\mathbf{P}$ can be used for the reparameterisation of the path. In practise, we have not encountered any difficulties related to this issue.

Here, we briefly describe the modifications required when the ℓ^2-metric is replaced with the metric $\mathrm{d}_{\mathbf{P}}$:

After accepting an optimisation step k of Eq. (4.10) the following steps are performed.

1. Compute the relative distances $\mathrm{d}_{\mathbf{P}}(\boldsymbol{x}_n^k, \boldsymbol{x}_{n-1}^k)$ between the images $\{\boldsymbol{x}_n^k\}_n$, for all $n = 2, \ldots, N$.

2. Define

$$s_1 = 0, \tag{4.14}$$
$$s_n = \frac{\sum_{m=2}^{n} \mathrm{d}_{\mathbf{P}}(\boldsymbol{x}_m^k, \boldsymbol{x}_{m-1}^k)}{\sum_{m=2}^{N} \mathrm{d}_{\mathbf{P}}(\boldsymbol{x}_m^k, \boldsymbol{x}_{m-1}^k)}, \quad \text{for } n = 2, \ldots, M.$$

3. Use cubic spline interpolation of $\{s_n, \boldsymbol{x}_n^k\}_{n=1}^N$ (refer to Sec. 2.7 for more details) to obtain $\boldsymbol{x}^k(s) : [0,1] \to \mathbb{R}^N$.

4. The new images are then given by

$$\boldsymbol{x}_n^k = \boldsymbol{x}^k\big(\tfrac{n-1}{N-1}\big), \qquad n = 1, \ldots, N. \tag{4.15}$$

This algorithm does *not* ensure that images will be equidistributed according to $\mathrm{d}_{\mathbf{P}}$. However it does ensure that images remain bounded away from one another, which is the key property required for the string method.

4.2.3 Preconditioned NEB method

An entirely analogous argument yields the preconditioned NEB method,

$$\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k + \alpha^k[-\nabla^{\perp} V_{\mathrm{P}}(\boldsymbol{x}_n^k) + (\boldsymbol{\eta}_{\mathrm{neb,P}})_n^k], \tag{4.16}$$

where

$$(\boldsymbol{\eta}_{\mathrm{neb,P}})_n^k = \kappa \left((\boldsymbol{x}_n^k)'' \cdot \mathbf{P}_n^k \frac{(\boldsymbol{x}_n^k)'}{\|(\boldsymbol{x}_n^k)'\|_{\mathbf{P}_n^k}} \right) \frac{(\boldsymbol{x}_n^k)'}{\|(\boldsymbol{x}_n^k)'\|_{\mathbf{P}_n^k}}. \tag{4.17}$$

4.3 Scheme Summary

To summarise, we propose the following preconditioned updating relations,

$$\boldsymbol{x}_n^{k+1} = \boldsymbol{x}_n^k + \alpha^k \left[-\nabla^{\perp} V_{\mathrm{P}}(\boldsymbol{x}_n^k) + (\boldsymbol{\eta}_{\mathrm{P}})_n^k\right], \tag{4.18}$$

where, in analogy to our earlier notation, $(\boldsymbol{\eta}_{\mathrm{P}})_n^k = (\boldsymbol{\eta}_{\mathrm{neb,P}})_n^k$ from Eq. (4.17) for NEB and $(\boldsymbol{\eta}_{\mathrm{P}})_n^k = 0$ for the string method, where additionally the redistri-

bution described in Sec. 4.2.2 follows this step.

The above class of preconditioning schemes disregards the interactions between images and therefore, the preconditioner aids the convergence of the path only in the transverse direction. This is justified when the main source of ill-conditioning is due to the potential energy landscape, which is the case when only few images are used as is often done in practise. For example for a complex model such as DFT using a large number of images is computationally expensive. Typically a number of images around 10 is used, although often much fewer images are used.

4.4 Geometry Optimisation Peconditioners

As we have previously mentioned, in our preconditioning scheme we use geometry optimisation preconditioners. Before we proceed to our results, we review in this section two particular geometry optimisation preconditioners which we will be using.

4.4.1 Exponential preconditioner

The *exponential preconditioner* (EXP) introduced in Packwood *et al.* [71] was developed to treat general condensed phase atomistic systems, by utilising bond-connectivity information. The preconditioner $\mathbf{P}_{\mathrm{EXP}}$ is defined in terms of the parameters r_{cut}, r_{nn}, A and μ, by

$$c_{ij} = \exp\bigg(- A\Big(\frac{r_{ij}}{r_{\mathrm{nn}}} - 1\Big)\bigg), \tag{4.19}$$

$$(\mathbf{P}_{\mathrm{EXP}})_{ij} = \begin{cases} -\mu c_{ij}, \text{ if } |r_{ij}| < r_{\mathrm{cut}} \\ 0, \text{ if } |r_{ij}| \geq r_{\mathrm{cut}} \end{cases}, \tag{4.20}$$

$$(\mathbf{P}_{\mathrm{EXP}})_{ii} = -\sum_{j \neq i} (P_{\mathrm{EXP}})_{ij}, \tag{4.21}$$

where i and j index the degrees of freedom of the system. The nearest neighbour distance r_{nn} is evaluated as the maximum of the nearest bond lengths. The value of A is chosen to be sufficiently large so that the contribution from nearest neighbours dominates, but small enough so that $\mathbf{P}$ is not sensitive to

small changes of the configuration. In [71], $A = 3.0$ gives good performance. The cutoff radius r_{cut} needs to be larger than r_{nn} and μ is chosen so that when used with the LBFGS algorithm (recall from Sec. 2.5.3), a unit step length can be chosen by default.

4.4.2 Force Field preconditioner

A class of preconditioners, that of the *force field* preconditioners (FF) is proposed in Mones *et al.* [2]. Such a preconditioner uses information from molecular mechanical models. Starting from a molecular mechanical potential

$$V_{\text{FF}} = \sum_{\alpha} V_{\alpha}(\xi_{\alpha}(\boldsymbol{x})) \tag{4.22}$$

which is expressed as the sum of interactions of short-range bonds over internal coordinates ξ_{α}. The Hessian $\mathbf{H}_{\alpha}$ associated with V_{α} may be then decomposed as

$$\mathbf{H}_{\alpha} = \frac{\partial^2 V_{\alpha}}{\partial \boldsymbol{x}^2} = \frac{\partial \xi_{\alpha}}{\partial \boldsymbol{x}} \otimes \frac{\partial \xi_{\alpha}}{\partial \boldsymbol{x}} \frac{\partial^2 V}{\partial \xi_{\alpha}^2} + \frac{\partial^2 \xi_{\alpha}}{\partial \boldsymbol{x}^2} \frac{\partial V_{\alpha}}{\partial \xi_{\alpha}} \tag{4.23}$$

where the first part is positive definite if V_{α} is quadratic. The preconditioner $\mathbf{P}_{\text{FF}}$ is constructed by summing over all α and for each contribution retaining only the first term of $\mathbf{H}_{\alpha}$ while simultaneously enforcing positive definiteness, according to

$$\mathbf{P}_{\text{FF}} = \sum_{\alpha} \frac{\partial^2 V_{\alpha}}{\partial \boldsymbol{x}^2} = \frac{\partial \xi_{\alpha}}{\partial \boldsymbol{x}} \otimes \frac{\partial \xi_{\alpha}}{\partial \boldsymbol{x}} \left| \frac{\partial^2 V}{\partial \xi_{\alpha}^2} \right|. \tag{4.24}$$

For crystal structures, an improved preconditioner considering the effects of intermolecular interactions may be attained by combining the EXP and FF preconditioners as described in [2, p. 4]. The only contributions of $\mathbf{P}_{\text{EXP}}$ kept, are the ones corresponding to zeros of the $\mathbf{P}_{\text{FF}}$. Then we can express

$$\mathbf{P}_{\text{FF+EXP}} = (1 - \lambda)\mathbf{P}_{\text{FF}} + \lambda \mathbf{P}_{\text{Exp}}. \tag{4.25}$$

The parameters r_{cut}, r_{nn} and A of $\mathbf{P}_{\text{EXP}}$ are chosen as described in [71] as previously mentioned in Sec. 4.4.1, whereas μ and λ are chosen by fitting $\mathbf{P}_{\text{Exp}}$ to $\mathbf{P}_{\text{FF}}$ to ensure correct scaling.

In [2] it is also addressed that it might necessary to regularise the pre-

conditioner to treat degeneracies of the $\mathbf{P}_{\mathrm{FF}}$, according to

$$\mathbf{P}_{\mathrm{regural}} = \mathbf{P} + c\mathbf{I}, \tag{4.26}$$

where $\mathbf{P}$ is either $\mathbf{P}_{\mathrm{FF}}$ or $\mathbf{P}_{\mathrm{FF+EXP}}$ and $c = 0.1,\ 1.0$ eV/Å^2 are typical values for c.

4.5 Results

We tested our preconditioning scheme for a variety of examples. First, we looked at examples using interatomic potentials which are not the main target, as these are typically fast models and constructing a preconditioner may not be computationally efficient relative to force evaluations and is therefore redundant. The preconditioner is meant to aid more difficult systems where the computational cost of the preconditioner is negligible in comparison to that of the force call and force preconditioning, such as a DFT model. However, the interatomic potential examples, demonstrate how the number of force evaluations can be reduced with the use of the preconditioner, in the situation where the "ideal" preconditioner closely describing the Hessian matrix can be used, to enable comparison with the more realistic DFT case which follows and where the exact preconditioner is not available.

We note that our described implementation updates the preconditioner after each iteration. In fact, the preconditioner can be efficiently re-used from one step to the next providing changes in the geometry are small. Further fine-tuning the preconditioner implementation and application, for example avoiding to update the preconditioner after each iteration, one would still obtain significant practical speed-ups for severely ill-conditioned cases even in the interatomic potentials case.

We then compare with a DFT model to confirm our earlier results. In the following tables we compare the number of force evaluations per image needed to converge to "coarse" and "fine" target accuracies (maximum force less than 10^{-1} eV/Å and 10^{-3} eV/Å, respectively) using unpreconditioned and preconditioned schemes with either static or adaptive *ode12r* step selection.

The criterion for convergence is the magnitude of the residual error

$$R^{k+1} = \max_n \left\| \mathbf{P}_n^k \left\{ \nabla^\perp V_{\mathrm{P}}(\boldsymbol{x}_n^k) \right\} \right\|_\infty. \tag{4.27}$$

4.5.1 Vacancy Migration

First we consider the diffusion of a vacancy in a two dimensional 60-atom triangular lattice governed by a Lennard-Jones potential, described earlier in Sec.2.3.1, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. The vacancy is located at the centre of the cell initially and migrates in the y direction by one lattice spacing. Periodic boundary conditions are imposed in the x and y directions.

Figure 4.4 shows the convergence of the string and NEB methods, as a function of force calls per image. Table 4.1 summarises the number of force calls per image required for convergence to a low and higher accuracies. The EXP preconditioner described in Sec 4.4.1 with parameters $A = 3.0$ and $r_{\mathrm{cut}} = 2.5$, to treat the ill-conditioning of the system allowed convergence beyond the 10^{-3} tolerance, which the unpreconditioned case could not achieve within a reasonable number of iterations. This strengthens the claim we made in Sec 3.3.2 regarding the ill-conditioning of this system. For the results of Table 4.1 and Fig 4.4 for the unpreconditioned case and the *ode12r* time stepping for the string method we chose to use $rtol = 0.01$ and $atol/rtol = 0.01$, which offered a good convergence rate up to an accuracy of $\sim 3 \times 10^{-2}$. The absolute differences $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_\infty$ of the positions of any two converged paths at the images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ nearest to the saddles, with and without preconditioning were of the order of 8×10^{-3}.

We study the number of evaluations until convergence for varying pairs ($atol$, $rtol$) of the *ode12r* time stepping for the string method and we plot the results in Fig. 4.5. The white coloured tiles identify the values of $atol$ and $rtol$ for which the method did not converge within a reasonable time. The target tolerance for the upreconditioned case was set to an accuracy of 8×10^{-2}, whereas for the preconditioned case a tolerance of 10^{-3} was pursued. The unpreconditioned method fails in general to converge within a reasonable time unless both $atol$ and $rtol$ are small and $atol \lesssim 10^{-2} rtol$. Further, as mentioned earlier for the results of Table 4.1, converging to an accuracy of 10^{-3}, proved to be impossible. This is due to severe ill-conditioning. The right hand panel

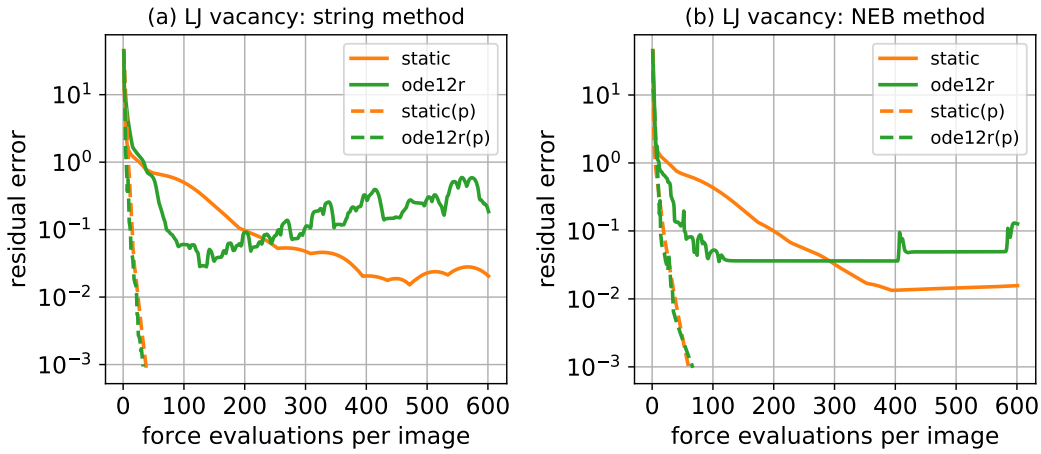


Figure 4.4: Convergence rate for the (a) string and (b) NEB methods for a vacancy migration in a 60-atom 2D cell modelled with the Lennard-Jones potential. The path consisted of 9 images and the static and *ode12r* step length selection schemes were used.

of Fig. 4.5 for the preconditioned results strengthens this claim as it is clear that the preconditioner gives fast convergence and the method is robust with respect to the choices of *atol* and *rtol*.

Next, we considered a three dimensional system containing a vacancy, specifically a 107-atom Cu FCC supercell in a fixed cell with periodic boundary conditions. Interactions were modelled with a Morse potential with parameters $A = 4.0$, $\epsilon = 1.0$ and nearest neighbour distance $r_0 = 2.55$ Å (refer to Sec. 2.3.1). The EXP preconditioner was used with parameters $A = 3.0$, $r_{\text{cut}} = 2.2$ and $r_0 = 5.62$ Å. Table 4.2 shows the number of force evaluations per image needed for convergence to two preset tolerance limits. This example demonstrates how the *ode12r* solver can aid the performance of the string and NEB methods if a static step is not suitable. Preconditioning gave almost a 2-fold speedup for the higher accuracy results, but no improvement for the lower accuracy. The absolute differences of the positions of the converged paths at the saddle, as done before, were well below 3×10^{-14}Å.

A 53-atom W BCC supercell modelled with the EAM4 potential described in Sec. 2.3.1 was examined as well. Periodic boundary conditions were imposed. Preconditioning was performed with the use of the FF preconditioner (Sec. 4.4.2). This yields up to 6 times faster convergence for higher accuracies as shown in Table 4.3. The absolute differences of the positions of

Lennard - Jones 2D Vacancy

Step selection	static		*ode12r* solver	
Tol	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	197	*	52	*
String (p)	16	38	12	33
NEB	200	*	53	*
NEB (p)	19	60	14	67

Table 4.1: Number of force evaluations per image required by the string and NEB methods to converge the vacancy migration MEP in a 9 image path of a 60-atom 2D cell modelled with a Lennard-Jones potential, with either the static or *ode12r* step length selection methods. In the cases marked *, the algorithm did not converge within a reasonable number of iterations.

Vacancy in Cu supercell

Step selection	static		*ode12r* solver	
Tol / eV/Å	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	8	74	8	41
String (p)	7	38	8	21
NEB	8	57	8	27
NEB (p)	7	37	8	19

Table 4.2: Force evaluations per image needed for the string and NEB methods for the migration of a vacancy in a 107-atom Cu fcc supercell modelled by a Morse potential. The MEP was discretised with 5 images.

the converged paths at the saddle, were well below 5×10^{-9}Å.

We studied the same 53-atom W vacancy system with DFT, as implemented in the CASTEP [44] software. The exchange correlation functional was approximated by PBE GGA [42], with a planewave energy cut-off of 500 eV and a $2 \times 2 \times 2$ Monkhorst-Pack grid to sample the Brillouin zone (Sec. 2.3.1). Step selection with *ode12r* step and static step selection schemes were studied. A regularised FF preconditioner based on the EAM4 Hessian was used, with parameters $c = 0.05$ and $\lambda = 0.4$ (refer to Sec. 4.4.2).

The path is made up of 5 images and traversing the path in subsequent iterations of the NEB and string methods was performed in an alternating order, allowing efficient reuse of previous electronic structure data to start the next optimisation step.

Unlike the EAM case above, the preconditioner we used for the DFT

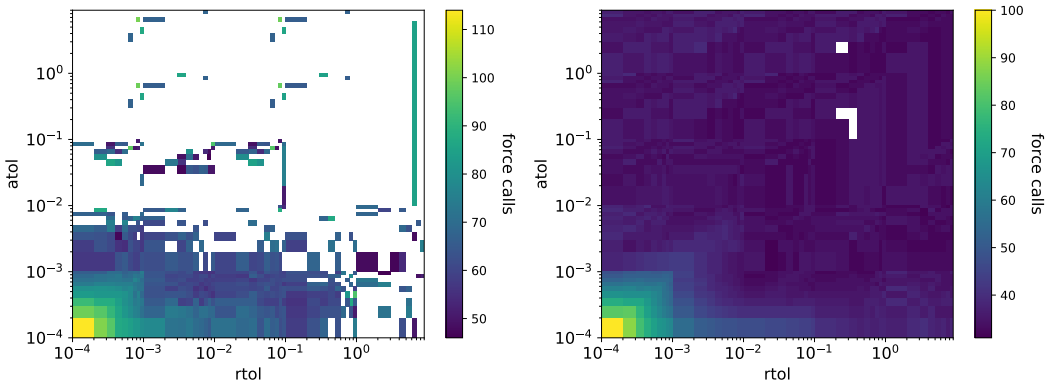


Figure 4.5: Convergence rate for the string method for a vacancy migration in a 60-atom 2D cell modelled with the Lennard-Jones potential. The path consisted of 9 images and the static and *ode12r* step length selection schemes were used. A map is constructed studying pairs of *atol* and *rtol*. White regions indicate cases where the string method did not converge within reasonable time.

model does not describe the potential energy surface of the DFT model exactly, but nevertheless gives a speed-up of a factor of 2 for an accuracy of $\sim 10^{-2}$ eV/Å and furthermore allows accuracies of the order of $\sim 10^{-3}$ eV/Å to be achieved, in contrast to the unpreconditioned case, as shown in Figs. 4.6 and 4.7. The results of Table 4.3 suggest that constructing a better preconditioner would improve these results further. Notice further that the number of force evaluations needed for convergence and the time needed for convergence are in agreement (by comparison of the upper and lower panes of Figs. 4.6 and 4.7), confirming that the computational cost of constructing the preconditioner model is negligible compared to the cost of computing DFT forces, justifying our earlier assumptions. We note that the gain of preconditioning would be expected to further increase with system size [71]. The maximum absolute differences of the positions of the converged paths at the saddle were of the order of 1×10^{-4}Å.

4.5.2 Screw Dislocation

In the final example we study a $\frac{1}{2}\langle 111 \rangle$ screw dislocation in a 562-atom W bcc structure confined in a cylinder of radius equal to 20 Å and surrounded

Vacancy in W supercell				
Step selection	static		ode12r solver	
Tol / eV/Å	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	7	77	7	49
String (p)	5	12	5	9
NEB	8	58	7	35
NEB (p)	5	10	8	17

Table 4.3: Force evaluations per image needed for the string and NEB methods to converge the MEP for vacancy migration in a 53-atom W bcc supercell modelled by the EAM4 potential [1]. The path was discretised by 5 images and the preconditioner was constructed from the force field [2].

by an 11 Å cylindrical shell of clamped atoms, with periodic boundary conditions along the dislocation line (z) direction (see Sec. 2.3.2). The system is simulated with the same EAM4 potential. The dislocation advances by one glide step. Table 4.4 shows the computational costs for converging the MEP with the NEB and string methods, using either static or *ode12r* step length selection. A force field preconditioner built from the EAM4 potential was used for geometry optimisation.

Screw Dislocation				
step selection	static		ode12r solver	
Tol / eV/Å	10^{-1}	10^{-3}	10^{-1}	10^{-3}
String	40	272	14	124
String (p)	7	48	9	21
NEB	40	312	14	162
NEB (p)	7	47	7	21

Table 4.4: Computational cost for the NEB and string methods for a screw dislocation in a 562-atom W bcc cylinder simulated with the EAM4 Marinica potential [1]. The circular boundary is fixed at a radius of R = 20Å. Periodic boundary conditions were imposed in the z direction. The path was discretised by 9 points.

Upon preconditioning, we observed a 5-fold speed up for the static case for low accuracies but only a 2-fold speedup for the *ode12r* case. For a higher accuracy, a speedup of a factor of 6 was observed and there was a speedup of a factor of at least 2 from using the *ode12r* step selection over the static step selection for both the unpreconditioned and preconditioned cases. This

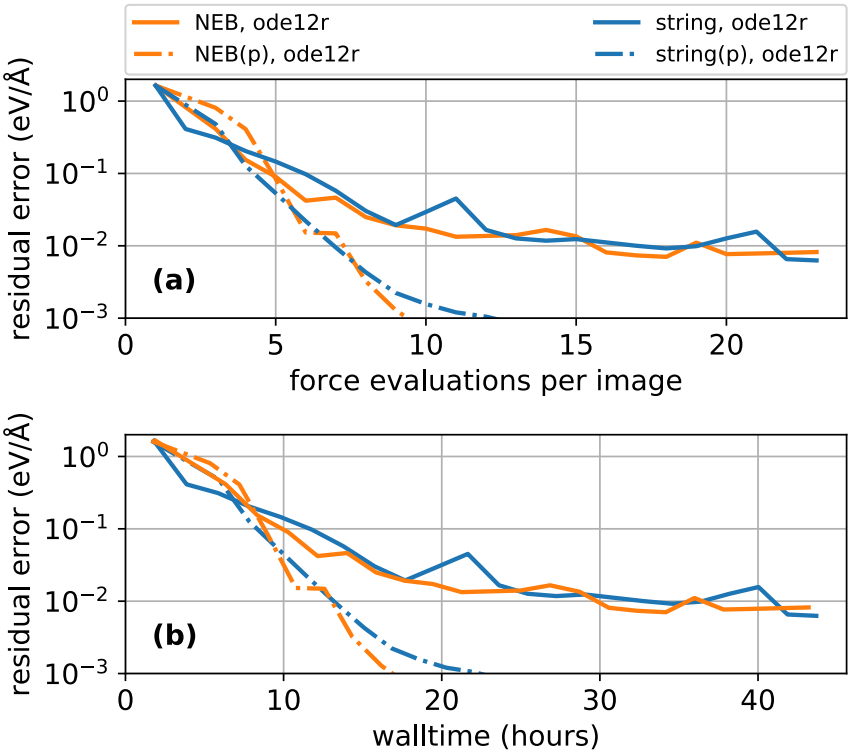


Figure 4.6: Convergence of the string and NEB methods with adaptive *ode12r* step selection, with and without preconditioner for a 53-atom bcc W supercell containing a vacancy and modelled with DFT. The upper panel (a) shows the error as a function of the number of force evaluations per image and the lower (b) as a function of the time required to converge. Time stepping with *ode12r* was used with a path of 5 images. Comparison shows that constructing and evaluating the preconditioner is negligible compared to the cost of force computation.

indicates that the fitted static step is only suitable in the pre-asymptotic regime and a larger step size is suitable in the asymptotic regime, showcasing the advantages of using the adaptive *ode12r* scheme over the hand-tuned static step. The absolute differences of the positions of the converged paths at the saddle were below 2×10^{-3}Å.

We investigated this system further, focussing on the NEB implementation to allow comparison with the widely used LBFGS [54] (see Sec. 2.5.3) optimisation algorithm, which can be used with the NEB implementation [14] in the Atomic Simulation Environment (ASE) [3]. This required fixing the endpoints of the path at the minima as is done in the ASE code. The compar-

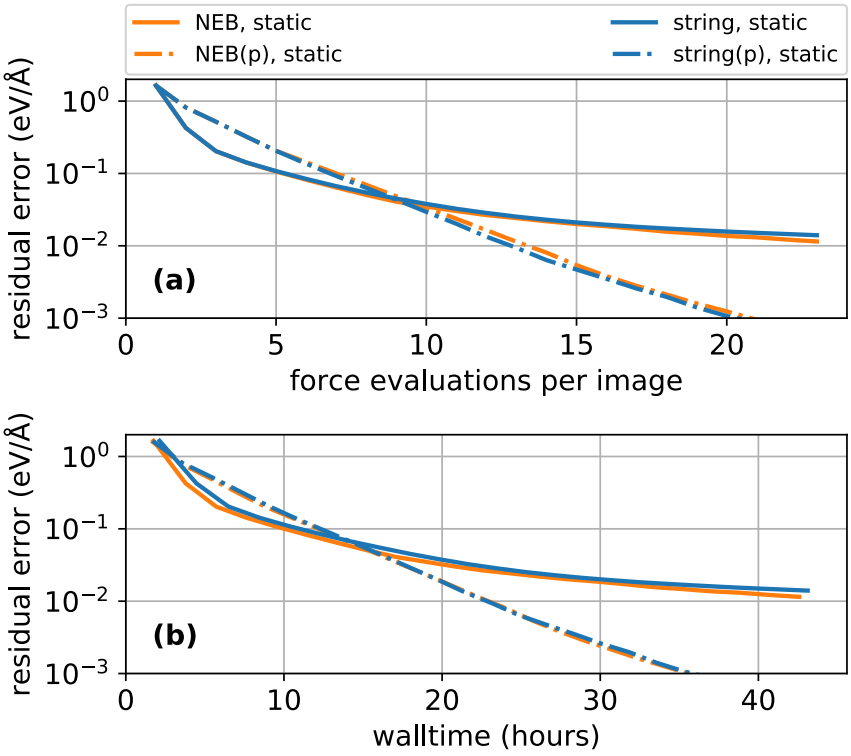


Figure 4.7: Convergence of the static string and NEB methods with and without preconditioner for a 53-atom W bcc supercell containing a vacancy and modelled with DFT. The upper panel (a) shows the error as a function of the number of force evaluations per image and the lower panel (b) shows the error as a function of the time required to converge. The static time step was chosen by extrapolating the *ode12r* data. The path was discretised by 5 images.

ison was carried out on systems of two sizes. An FF preconditioner was used as before for the preconditioned cases. Figure 4.8 shows the convergence rate of the various NEB schemes for a radius of 20 Å in the upper panel (a) and for a radius of 40 Å in the lower panel (b). Note that although LBFGS gave good convergence in the unpreconditioned case, it lacks robustness. This is because the force field of the NEB algorithm is not conservative, violating one of LBFGS's assumptions. LBFGS constructs a Hessian matrix corresponding to a scalar field (see Sec. 2.5.3), failing to capture the effects of the transport terms of the NEB force field. Moreover, the lack of the energy function prevents the use of line search, required to ensure the method's stability; in the ASE LBFGS implementation a heuristic is instead used to impose a maximum step length of 0.04 Å. Furthermore, it should be noted that because our

preconditioning scheme does not treat the longitudinal force components, it is inappropriate for us to use it together with the LBFGS method for MEP finding methods.

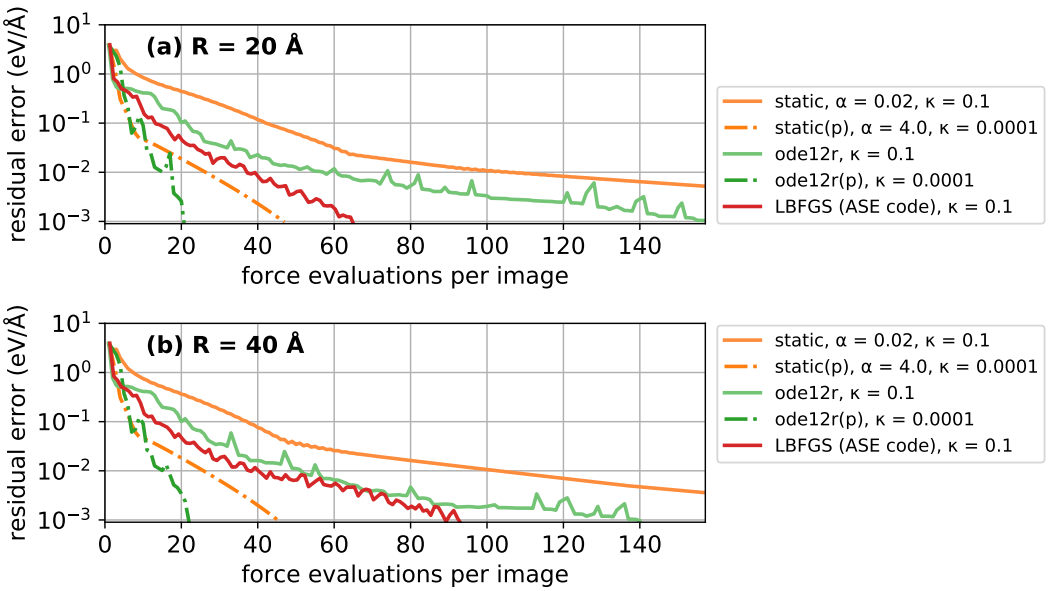


Figure 4.8: Convergence of NEB variants for a screw dislocation in a 562-atom W bcc cylindrical structure (a) and a 1489-atom W bcc cylindrical structure (b) modelled with the EAM4 Marinica potential [1]. Atoms outside outer radii of R = 20Å and R = 40Å respectively were clamped, with periodic boundary conditions along the dislocation line. The path was discretised with 7 images (excluding the minima at each end, which were held fixed). The horizontal axis of the plots was cut after 160 force evaluations per image to focus on the performance of the preconditioned schemes. The static unpreconditioned NEB method converged after 312 force evaluations per image for the R = 20Å case and after 343 force evaluations per image for the R = 40Å case.

4.5.3 Hybrid models

We now discuss hybrid models. Consider for example the following: On the atomistic scale, accurate analysis of bond breaking and bond formations, charge transfer and electron excitation requires a quantum mechanical treatment. However, the computations involved are expensive dictating restriction on the size of the system to a few hundred atoms [74]. For systems where long range interactions define their behaviour, and small regions in atomistic

regime can be affected over long distances from the behaviour of large collections of atoms or the converse, typical quantum mechanical (QM) approaches will miss important events whereas molecular simulations will give poor accuracy. A technique that may be adopted to treat such strongly multiscale systems and achieve accuracy comparable to an atomistic model, whilst benefiting from the computational efficiency of a molecular model (MM) is a hybrid model, see for example [74, 75, 76]. Long range interactions would be treated by molecular simulations whereas the details of the atomistic interactions in a region of interest are captured by the quantum mechanical treatment. Care needs to be taken however, so that the system's behaviour on a macroscopic scale is equivalent to a fully quantum mechanical model within a reasonable tolerance limit.

In choosing which regions of the system need a QM treatment and which regions should be modelled by MM, intuition suggests that localised topological defects require a QM approach, to capture the details of bond rearrangements owed to the strongly concentrated stress field. These, are atoms that 'trigger' a QM treatment, the active atoms. Neighbouring atoms should also be modelled accordingly. The aim is to obtain equivalent forces on each molecular cell as the total force obtained by the QM treatment $\mathbf{F}^i \approx \mathbf{F}^i_{\mathrm{QM}}$ for each atom i. In modelling the transition from the QM model to the MM one, in general two approaches are used: (i) coupling by total energy function considerations and (ii) QM and MM force coupling. The former approach uses a linear combination of the energies of both models, in such a way that the energy over the system is conserved. This approach suffers from the effect of ghost forces at the transition interface. In the latter case, we can no longer assume that the resulting forces correspond to a gradient of a potential with a closed form expression. To achieve momentum conservation in this case, the QM forces need to be shifted by a constant [74, 76]. Force coupling has the advantage of reducing boundary effects, particularly for covalent/metallic systems. For this reason, we are adopting the latter approach.

By construction, the string and NEB method are both expressed in terms of force calls alone - thus we are allowed to use models described by forces that do not correspond to a gradient of any potential, such as the QM/MM model described above. Our implementation of the preconditioned string and

NEB offers this feature as well.

To demonstrate that the proposed preconditioned schemes may be used in a hybrid model, we consider a $\frac{1}{2}\langle 111\rangle$ screw dislocation in a 1489-atom W BCC structure confined in a cylinder of radius equal to 40 Å and surrounded by an 11 Å cylindrical shell of clamped atoms, with periodic boundary conditions along the dislocation line (z) direction (see Sec. 2.3.2). We use a coupled EAM2-EAM4 model [48], where EAM2 and EAM4 refer to the Marinica EAM potentials [1]. In particular, a cylinder of radius equal to 18 Å (containing the core of the screw dislocation) is treated with the EAM2 potential, surrounded by a cylindrical shell of radius 12 Å acting as a buffer region, where the EAM2-EAM4 force matching takes place, and the remaining outer shell of the system is treated with the EAM4 potential. Preconditioning was performed with an FF preconditioner based on the EAM4 Hessian (Sec. 4.4.2).

We use the string and NEB methods with a 9-image path and fixed ends. We plot the convergence results in Fig. 4.9a and 4.9b. We observed a 6-fold speedup from the use of the preconditioner. Figures 4.9c and 4.9d show the energy profiles of the converged paths for the string and NEB methods respectively demonstrating that the paths found with the preconditioned methods agree with the paths found from the unpreconditioned methods. Following this set-up one may use straightforwardly a QM treatment in place of the EAM2 model used here.

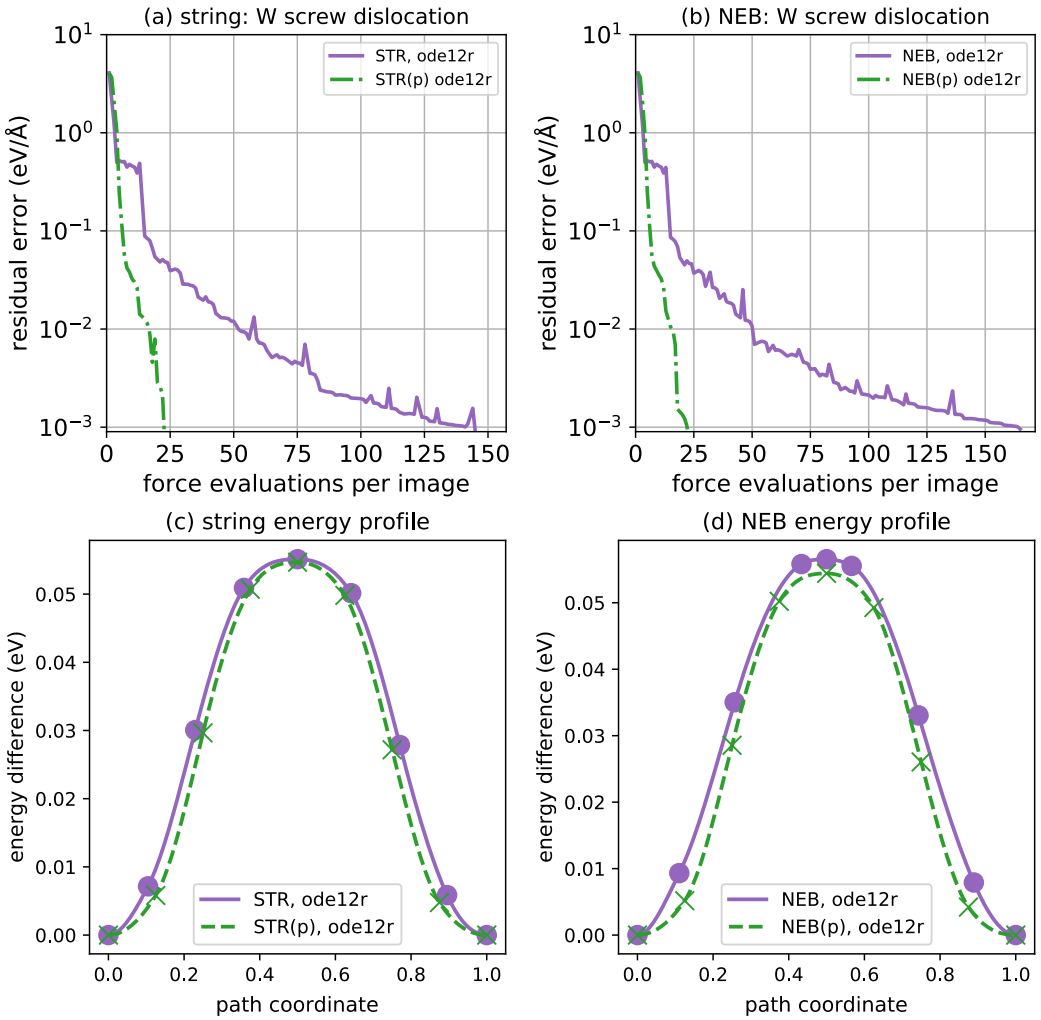


Figure 4.9: Convergence of the (a) string and (b) NEB methods for a screw dislocation in a 1489-atom W bcc structure confined in a cylinder of radius equal to R = 40Å and periodic boundary conditions along the dislocation line. Atoms outside outer radius of R were clamped. The system is modelled with a coupled EAM2-EAM4 potentials. (c) and (d) show the energy profiles of the MEPs found by the string and NEB methods respectively.

4.6 Conclusions

In this section, we explored the convergence of MEP finding methods in ill-conditioned potentials. We have demonstrated that in the presence of ill-conditioning, techniques such as the NEB and the string method can exhibit slow convergence rates and we proposed a preconditioning scheme to treat

these cases, to allow faster convergence to the minimum energy path.

The proposed scheme treats each image along the path independently, using preconditioners from geometry optimisation and thus only addressing ill-conditioning of the energy landscape in directions transverse to the path. Further, we combined the preconditioning scheme with the adaptive time-stepping scheme *ode12r* of Ch. 3 to improve the robustness of the scheme.

We observed that our new scheme gives a significant speedup and improved robustness over currently used approaches for a range of systems using both force fields and DFT. Speedups of factors as high as 6 were observed for preconditioners which closely describe the curvature of the potentials Moreover, it allows higher accuracies to be reached than existing methods.

However, our preconditioning scheme targets transverse ill-conditioning only. The longitudinal terms, (e.g. the NEB spring interactions) are unaffected by the preconditioner, suggesting that our scheme provides a baseline for further improvements.

Results from this chapter were published in [70]. An open source prototype implementation of our technique is available at `https://github.com/cortner/SaddleSearch.jl`.

Chapter 5

Momentum Descent

5.1 Introduction

In Sec. 2.5.6 we briefly introduced momentum descent methods, also known as accelerated gradient descent. This is a family of 2-step methods, which, similarly to steepest descent, may be used to tackle problems of minimising a differentiable functional, say $V(\boldsymbol{x})$, as long as the problem may be reduced to solving $\nabla V(\boldsymbol{x}) = 0$ with respect to $\boldsymbol{x}$ [57]. In fact, the method is more general, it may be used to solve $\mathbf{F}(\boldsymbol{x}) = 0$ provided that the solution $\boldsymbol{x}^*$ is a stable equilibrium of the dynamical system $\dot{\boldsymbol{x}} = \mathbf{F}(\boldsymbol{x})$ and that $\mathbf{F}$ is reasonably well-behaved. Usually one assumes that $\mathbf{F}$ is differentiable and that the solution to $\mathbf{F}(\boldsymbol{x}) = 0$ exists.

A 2-step method is one where the iterative scheme will use the current and the previous approximations, $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k-1}$, in order to make a new prediction $\boldsymbol{x}^{k+1}$. The idea is to incorporate some form of momentum into the steepest descent method, in order to accelerate convergence. As mentioned in Sec. 2.5.6, the first example of a momentum descent method in literature is Polyak's heavy ball method [57] which incorporates the momentum term

$$\beta(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}), \tag{5.1}$$

into Eq. (2.56) which we restate below

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha\mathbf{F}(\boldsymbol{x}^k) + \beta(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}), \tag{2.56}$$

In the above expression, α is the step size and β is a dimensionless parameter measuring momentum taken in the interval $(0, 1)$. As the name implies, the scheme is to be thought of as simulating the behaviour of a heavy ball rolling down a 'hill' defined by the potential landscape of V. The 'heaviness' is owed to the added inertia coming from the momentum term.

Of particular interest to us is the local convergence rate of this method in terms of the condition number κ of the Jacobian $\nabla\mathbf{F}$. Suppose that $\mathbf{F}$ is conservative, $\mathbf{F} = \nabla V$, with associated V the quadratic form

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot \mathbf{Q}\boldsymbol{x} - \boldsymbol{d} \cdot \boldsymbol{x}, \tag{5.2}$$

where $\mathbf{Q} \in \mathbb{R}^{M\times M}$ is symmetric and positive definite, and $\boldsymbol{d} \in \mathbb{R}^M$, then in this case Polyak [57] proves that if α and β are chosen optimally,

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}} \leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \left(\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_{\mathbf{Q}}^2 + \|\boldsymbol{x}^1 - \boldsymbol{x}^*\|_{\mathbf{Q}}^2\right)^{1/2}. \tag{5.3}$$

which is to be compared to the analogous relation for the gradient descent method given by (2.63) and which we restate here for completeness,

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}} \leq \left(\frac{\kappa-1}{\kappa+1}\right)^k \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_{\mathbf{Q}}, \tag{2.63}$$

but more importantly to be compared to the convergence rate of the conjugate gradient method (2.69) which we re-state below

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{Q}} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_{\mathbf{Q}}. \tag{2.69}$$

We emphasise that these convergence results are only valid in an optimal scenario where V has the form (5.2). More on optimality will follow later on in this chapter. Another important scheme is that of Nesterov's accelerated gradient descent [58], which was also introduced in Sec. 2.5.6 and is described

by Eq. (2.57) which we restate below

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{v}^k - \alpha \nabla V(\boldsymbol{v}^k), \\ \boldsymbol{v}^{k+1} &= \boldsymbol{x}^{k+1} + \frac{k-1}{k+2}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k). \end{aligned} \tag{2.57}$$

and has $\beta = (k-1)/(k+2)$ monotonically increasing as the iterations progress.

The idea of momentum acceleration was revisited recently. For example Su *et al.* [77], proved that Nesterov's accelerated gradient descent is a discretisation of a second order ODE. Wibisono *et al.* [78] went beyond this observation, to show that all accelerated descent methods of form (2.57), are discretisations of equations of motion arising from a generalised Lagrangian action. In particular, for elliptic problems, the resulting equation of motion represents a damped, nonlinear, wave equation, whereas, a gradient descent equation is a discretisation of a nonlinear diffusion equation. This provided the framework to analyse momentum descent in the PDE-setting as in Calder *et al.* [79] and Benyamin *et al.* [80]

The results of Calder *et al.* [79] are of particular interest to us. They proved that the equation of motion,

$$\ddot{\boldsymbol{x}} + b\dot{\boldsymbol{x}} + \boldsymbol{L}\boldsymbol{x} = \boldsymbol{0}, \tag{5.4}$$

converges to the steady state solution $\boldsymbol{x}^*$ of $\nabla V(\boldsymbol{x}^*) = 0$ exponentially fast, where $\boldsymbol{L}$ is a linear, self-adjoint[1] and positive definite second order partial differential operator. On the basis of the above result, Benyamin *et al.* [80] provided the framework to extend these results to specific cases of non-linear $\boldsymbol{L}$. We present here a summary of the results of Calder *et al.* [79], but we adopt a different notation which is more appropriate in our setting. Consider the linear map,

$$\begin{aligned} \mathbf{F} : \mathbb{R}^M &\to \mathbb{R}^M \\ \boldsymbol{x} &\mapsto \boldsymbol{L}\boldsymbol{x}, \end{aligned} \tag{5.5}$$

[1] An operator $\boldsymbol{L}$ is self-adjoint if $\forall \boldsymbol{x}, \boldsymbol{y}$, the following is true $\boldsymbol{L}\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{L}\boldsymbol{y}$

so that equation (5.4) may be re-written as,

$$\ddot{\boldsymbol{x}} + b\dot{\boldsymbol{x}} + \mathbf{F}(\boldsymbol{x}) = \mathbf{0}. \tag{5.6}$$

We call $b > 0$ the momentum coefficient, which is not to be confused with the earlier momentum parameter of Polyak's heavy ball scheme.

The Fourier series of the solution $\boldsymbol{x}^*(\tau)$ of Eq. (5.6),

$$\boldsymbol{x}^*(\tau) = \sum_{m=1}^{M} x_m(\tau)\boldsymbol{v}_m, \tag{5.7}$$

is expressed in terms of the eigenfunctions $\{\boldsymbol{v}_m\}_{m=1}^M$ of the Jacobian matrix $\boldsymbol{L} = \nabla\mathbf{F}(\boldsymbol{x})$ of $\mathbf{F}$ with eigenvalues $\{\lambda_m : 0 < \lambda_i < \lambda_{i+1}\, \forall i = 1, \ldots, \infty\}_{m=1}^M$ [79]. Substituting the Fourier expansion in Eq. (5.6), we find the general solution for the coefficients $x_m(\tau)$ to be

$$x_m(\tau) = Ae^{r_m^+\tau} + Be^{r_m^-\tau}, \tag{5.8}$$

where

$$r_m^{\pm} = -\frac{b}{2} \pm \frac{1}{2}\sqrt{b^2 - 4\lambda_m}. \tag{5.9}$$

To ensure that the Fourier coefficients decay, and that the method will converge as $\tau \to \infty$, it is necessary that the real part of both $r_m^{\pm}$ is negative which is achieved if

$$b^2 - 4\lambda_m \leq 0, \quad b > 0. \tag{5.10}$$

Then, the Fourier coefficients decay at a rate $\exp(-b\tau/2)$ which is optimal when

$$b = 2\sqrt{\lambda_1}, \tag{5.11}$$

in the sense that the $\exp(r_m^{\pm}\tau)$ will decay the fastest. Therefore, the convergence rate of $\boldsymbol{x}(\tau)$ to the true solution $\boldsymbol{x}^*$ as $\tau \to \infty$ is given by

$$|\boldsymbol{x}(\tau) - \boldsymbol{x}^*| \leq C\exp(-2\sqrt{\lambda_1}\tau), \tag{5.12}$$

for some constant $C > 0$ determined by the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$. This concludes the short review of the results from Calder *et al.* [79].

We want to incorporate the momentum descent scheme into geometry optimisation and saddle search methods. It is important to note that in the proof of [79], it was assumed that all eigenvalues of $\nabla \mathbf{F}(\boldsymbol{x})$ are real and positive, a condition that is only valid near an equilibrium. In the case of saddle point search methods, this condition is less strict. Due to the transport terms, which we will see later on in Sec. 5.5, near the equilibrium one can only guarantee that the real part of the eigenvalues are positive.

In this chapter we investigate the stability of the method for the case where the Jacobian of $\mathbf{F}$ has complex eigenvalues with positive real part and we find the optimal damping coefficient b in this case. We proceed by proposing three discretisation schemes and we investigate each of them in conjunction with the string, NEB and dimer methods. We conclude by demonstrating how momentum descent with carefully chosen values for the damping coefficient and time step parameters improves convergence in a static setting (the time step is not adaptively chosen).

Note that the finding of this chapter go beyond the work of Probert [81] which focuses only on optimisation, and where preconditioning is realised as a rescaling of the forcing term acting on the decoupled eigenmodes.

5.2 Stability and optimal momentum coefficient

Using (5.7) to decouple equation (5.6), leads to one-dimensional problems of the form

$$\begin{cases} \ddot{x} + b\dot{x} + \lambda x = 0 \\ x = x_0 \text{ and } \dot{x} = \dot{x}_0 \qquad \text{at } \tau = 0, \end{cases} \tag{5.13}$$

where we dropped the subscripts m for simplification of notation.

We now study in detail the convergence of (5.13) to equilibrium and the optimisation of the damping parameter b, in the case when λ is complex. Let us suppose that we are in the regime where $\mathrm{Re}(\lambda) > 0$, i.e $x(t)$ is an attracting trajectory and that $\mathrm{Im}(\lambda) \neq 0$. Recall that the trajectory $x(t)$ in Eq.(5.13) converges to equilibrium if and only if the real part of both $r^{\pm}$ defined in (5.9) are negative.

For a fixed λ we consider $r^{\pm}$ as a function of b and define

$$\begin{aligned} g^{\pm}(b) :=& -2\mathrm{Re}(r^{\pm}(b)) \\ =& b \mp \sqrt{\frac{\sqrt{(b^2-4\mathrm{Re}(\lambda))^2+16\mathrm{Im}(\lambda)^2}+b^2-4\mathrm{Re}(\lambda)}{2}}. \end{aligned} \tag{5.14}$$

Notice that

$$\sqrt{(b^2-4\mathrm{Re}(\lambda))^2+16\mathrm{Im}(\lambda)^2}+b^2-4\mathrm{Re}(\lambda) \geq |b^2-4\mathrm{Re}(\lambda)|+b^2-4\mathrm{Re}(\lambda) \tag{5.15}$$

with equality if and only if $\mathrm{Im}(\lambda) \equiv 0$ and $b < 2\mathrm{Re}(\lambda)$. But we are only considering the cases where $\mathrm{Im}(\lambda) \neq 0$ for the time being, thus,

$$\sqrt{(b^2-4\mathrm{Re}(\lambda))^2+16\mathrm{Im}(\lambda)^2}+b^2-4\mathrm{Re}(\lambda) > 0, \tag{5.16}$$

and it follows that $g^{\pm}$ have no points of discontinuity for $b > 0$.

We are left with identifying the values of $b > 0$, for which $g^{\pm}(b) > 0$ in order for $\exp(r^{\pm}\tau)$ to decay exponentially. It is trivial to see that $g^{-}(b) > 0$ is true for all $b > 0$. Further, from $b > 0$ and equation (5.16), it follows that

$$g^{-}(b) > g^{+}(b), \tag{5.17}$$

and therefore $\exp(r^{-}\tau)$ always decays faster than $\exp(r^{+}\tau)$.

Some elementary algebraic manipulations show that equating $g^{+}(b) = 0$ and requiring that $b > 0$ yields that g^{+} has a single point of intersection with the positive b-axis which is at

$$b_{\text{intersect}} = \frac{|\mathrm{Im}(\lambda)|}{\sqrt{\mathrm{Re}(\lambda)}}. \tag{5.18}$$

It would have been sufficient to determine the asymptotic behaviour of g^{+} in the neighbourhood of $b_{\text{intersect}}$ to decipher which values of b we need, however, it is more meaningful for us to understand the behaviour of g^{+} $\forall b, \lambda$. In fact we will check that g^{+} has a very simple shape, with a single maximum point which will give an analytic expression for the optimal b value.

For this reason we explore in Appendix A, the asymptotic behaviour of

g^+ as it approaches zero and infinity. From (A.2) in Appendix A, as $b \to 0^+$, the asymptotic behaviour of g^+ is

$$g^+(b) = -2\sqrt{2}\big(|\lambda| - \mathrm{Re}(\lambda)\big)^{\frac{1}{2}} + b + \mathcal{O}(b^2). \tag{5.19}$$

Therefore, g^+ behaves linearly, with a positive slope of 1, cutting the vertical axis below zero. Notice that indeed $|\lambda| - \mathrm{Re}(\lambda) > 0$, $\forall \lambda \in \mathbb{C}$ which follows trivially from Pythagoras' theorem confirming the expression is always real. Continuity of g^+ implies that

$$b > \frac{|\mathrm{Im}(\lambda)|}{\sqrt{\mathrm{Re}(\lambda)}} \Rightarrow g^+(b) > 0. \tag{5.20}$$

On the other hand, we see from (A.3) in Appendix A, that as $b \to \infty$, g^+ behaves like

$$g^+(b) = 2\mathrm{Re}(\lambda) b^{-1} + \mathcal{O}(b^{-3}). \tag{5.21}$$

Results (5.18), (5.19) and (5.21), together with the fact that g^+ is continuous, indicate the existence of a global maximum of g^+.

At this maximum, $\exp(r^+\tau)$ decays most rapidly. Recall from (5.17) that $\exp(r^-\tau)$ always decays faster than $\exp(r^+\tau)$. Thus, at the global maximum of $g^+(b)$, we achieve optimal convergence.

In Appendix A in Sec. A.3, we show that the fixed points of g^+ satisfy

$$b^2 - 4\mathrm{Re}(\lambda) = \beta \tag{5.22}$$

where β is the root of the cubic polynomial,

$$q(\beta) = 4\alpha\beta^3 - 3\gamma^2\beta^2 + 6\alpha\gamma^2\beta + (\alpha^2\gamma^2 - 4\gamma^4) = 0, \tag{5.23}$$

where we assigned the constants $\alpha := 4\mathrm{Re}(\lambda) > 0$ and $\gamma := 4\mathrm{Im}(\lambda) \neq 0$. In Sec. A.3 of Appendix A we also study the discriminant of q and we find that the discriminant satisfies

$$\Delta = -432\gamma^{10} - 1296\alpha^2\gamma^8 - 1296\alpha^4\gamma^6 - 432\alpha^6\gamma^4 < 0, \quad \forall \gamma \neq 0 \tag{5.24}$$

and thus the equation has a single real root, β_{opt}, which may be derived algebraically for example from Cardano's formula [82]. The other two roots are complex. Therefore, there is a single fixed point for $g^+(b)$ in $(0, \infty)$ given by

$$b_{\text{opt}} = \sqrt{\beta_{\text{opt}} + 4\text{Re}(\lambda)}. \tag{5.25}$$

Fig. 5.1 shows the general shape of $g^+(b)$ with the asymptotic behaviour near $b \to 0^+$ and $b \to \infty$ indicated in green and the maximum point marked with orange.

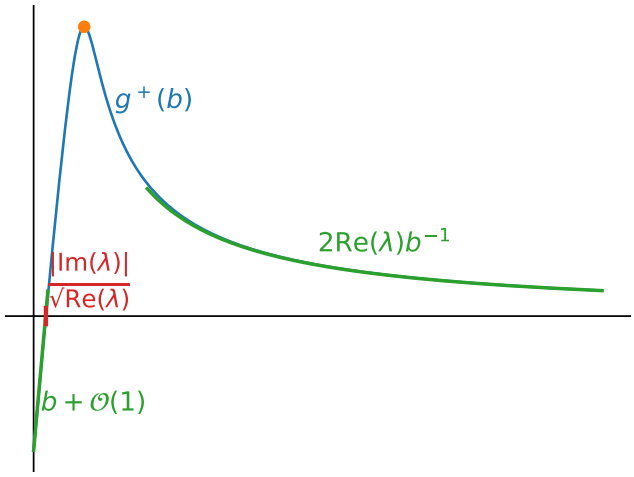


Figure 5.1: From the asymptotic behaviour of g^+ as $b \to 0^+$ and as $b \to \infty$ we sketch g^+. There is a maximum point, marked in orange. For this choice of b, x decays faster to the solution x^*.

We conclude this section by looking at the limit $\text{Im}(\lambda) \to 0$. From the general solution of a cubic polynomial [82] and (5.22), we can deduce that as $\text{Im}(\lambda) \to 0$, $\beta_{\text{opt}} = \mathcal{O}[\text{Im}(\lambda)^{2/3}]$ and b_{opt} reduces to the result from [79], that $b_{\text{opt}} = 2\sqrt{\text{Re}(\lambda)}$, as expected.

5.2.1 Summary

To summarise, we have shown that for the decoupled continuous momentum descent method, described by equation (5.13), along a direction corresponding

to an eigenvalue with positive real part, there is a unique choice for $b > 0$, offering optimal convergence speed. This value is given implicitly from relation (5.25) and the solutions of the cubic polynomial (5.23), which can be computed analytically from Cardano's formula [82] in $\sim \mathcal{O}(1)$ operations.

5.3 Numerical description of the momentum descent method

In this study, I will use three discretisation schemes of (5.4) resulting from utilising the *forwards*, *backwards* and *central* differences of $\dot{x}$. We discretise the pseudo-temporal coordinate t of x by points separated at intervals $\Delta\tau := h$ and we let $x^{(n)}$ approximate the value of x at the n^{th} discrete point $x^{(n)} \approx x(\tau = nh)$. In the present section we derive each discrete scheme explicitly and we analyse the stability of each of the schemes with respect to the eigen-spectrum λ and the time step h.

5.3.1 Discretising the momentum descent equation

We first look at the three aforementioned discretisations of the momentum descent method as they result from approximating the differential operators of (5.13) with finite differences.

Forward Difference Momentum Descent

Starting from equation (5.13) we approximate $\ddot{x}^{(n)}$ and $\dot{x}^{(n)}$ with a finite difference approximation using values of x evaluated at instances forwards in time. In particular, we obtain

$$\begin{aligned} x_{tt} + bx_t + \lambda x &= 0 \\ \frac{x^{(n+2)} - 2x^{(n+1)} + x^{(n)}}{h} + b\frac{x^{(n+1)} - x^{(n)}}{h} + \lambda x^{(n+1)} &= 0 \\ \Rightarrow x^{(n+2)} + x^{(n+1)}(hb - 2 + \lambda h^2) + (1 - hb)x^{(n)} &= 0 \\ \Rightarrow \boxed{x^{(n)} = (2 - hb - \lambda h^2)x^{(n-1)} - (1 - hb)x^{(n-2)},} & \end{aligned} \tag{5.26}$$

giving a discretised scheme for updating $x^{(n)}$ which we refer to as the *forward difference momentum descent*.

Backward Differences

Now we use backward differences for the differential operators to find the following relation

$$\begin{aligned} x_{tt} + bx_t + \lambda x &= 0 \\ \frac{x^{(n)} - 2x^{(n-1)} + x^{(n-2)}}{h} + b\frac{x^{(n)} - x^{(n-1)}}{h} + \lambda x^{(n-1)} &= 0 \\ \Rightarrow x^{(n)}(1+hb) + x^{(n-1)}(-2 - hb + \lambda h^2) + x^{(n-2)} &= 0 \\ \Rightarrow \boxed{x^{(n)} = \frac{2 + hb - \lambda h^2}{1+hb}x^{(n-1)} - \frac{1}{1+hb}x^{(n-2)}.} \end{aligned} \tag{5.27}$$

In similar fashion to earlier on, we refer to this scheme as the *backward difference momentum descent*.

Central Differences

Finally, we consider central differences to approximate the differential operators to get

$$\begin{aligned} x_{tt} + bx_t + \lambda x &= 0 \\ \frac{x^{(n+1)} - 2x^{(n)} + x^{(n-1)}}{h} + b\frac{x^{(n+1)} - x^{(n-1)}}{2h} + \lambda x^{(n)} &= 0 \\ \Rightarrow x^{(n+1)}(2+hb) + x^{(n)}(-4 + 2\lambda h^2) + (2 - hb)x^{(n-1)} &= 0 \\ \Rightarrow \boxed{x^{(n)} = \frac{2(2-\lambda h^2)}{2+hb}x^{(n-1)} - \frac{2-hb}{2+hb}x^{(n-2)},} \end{aligned} \tag{5.28}$$

and we refer to this scheme as the *central difference momentum descent*.

5.3.2 Stability regions

We call a scheme *stable* if in the limit $n \to \infty$ the approximate solution $x^{(n)}$ of respectively, expressions (5.26), (5.27) and (5.28), converges to zero. Fixing the choice of b, the series of estimates $x^{(n)}$ depend on the value of λ and the h

of choice. We will study the set of values λh^2 for which $\lim_{n\to\infty} x^{(n)}(\lambda, h) = 0$ and we denote this region by $\mathcal{D}$, that is

$$\mathcal{D} = \left\{ \lambda h^2 : \lambda h^2 \in \mathbb{C} \text{ and } \lim_{n\to\infty} x^{(n)}(\lambda, h) = 0 \in \mathbb{R} \right\}. \tag{5.29}$$

To get a sense of what values λh^2 will yield a stable method, we first consider a trivial generalisation of Calder and Yezzi's [79] result, $b = 2\sqrt{\mathrm{Re}(\lambda)}$, which ensures that b is real and we then proceed to the case where b is chosen optimally according to (5.25), $b = b_{\mathrm{opt}}$, as suggested by the results of Sec. 5.2. We refer to the former as *Calder's choice* and the latter as the *improved choice*.

The reason why (5.29) is a meaningful definition will become apparent if we consider the following. Provided that $\mathrm{Re}(\lambda) > 0$, there is a bijection

$$\lambda h^2 \leftrightarrow hb, \tag{5.30}$$

in that hb is uniquely determined from λh^2 if b is equal to either Calder's choice, or the improved choice. For Calder's choice, the bijection may be seen from $hb = 2\sqrt{\mathrm{Re}(\lambda h^2)}$ whereas for the improved choice, the bijection follows from $hg^{\pm}(b;\lambda) = g^{\pm}(hb;\lambda h^2)$. Thus, the definition of the stability region of each scheme as a region of the λh^2-complex plane is well defined.

All three schemes of Sec. 5.3.1 are of the general form

$$Ax^{(n)} + Bx^{(n-1)} + Cx^{(n-2)} = 0. \tag{5.31}$$

There is a general solution to (5.31) of the form $x^{(n)} = \omega^n$ and the two possible solutions for ω may be derived from the quadratic formula. Further, the $x^{(n)}$ converges to zero as $n \to \infty$ if and only if, $|\omega| < 1$. This will give us the stability condition for each of the three schemes. In particular, solving for ω with the respective coefficients of $x^{(n)}$, $x^{(n-1)}$ and $x^{(n-2)}$ from (5.26), (5.27), (5.28) will give the following three stability regions:

For the forward difference momentum descent scheme, the stability region is given by

$$\mathcal{D}_{FD} = \{\lambda h^2 : |2 - hb - \lambda h^2 \pm \sqrt{(hb)^2 + (\lambda h^2)^2 - 4\lambda h^2 + (\lambda h^2)(hb)}| < 2\}. \tag{5.32}$$

In the case of the backward difference momentum descent scheme, the stability region is

$$\mathcal{D}_{BD} = \left\{ \lambda h^2 : \left| \frac{2 + hb - \lambda h^2 \pm \sqrt{(2 + hb - \lambda h^2)^2 - 4(1 + hb)}}{1 + hb} \right| < 2 \right\}. \quad (5.33)$$

And finally, for the central difference momentum descent scheme, it is

$$\mathcal{D}_{CD} = \left\{ \lambda h^2 : \left| \frac{2 - \lambda h^2 \pm \sqrt{(2 - \lambda h^2)^2 - 4(1 - (hb)^2)}}{2 + hb} \right| < 1 \right\}. \quad (5.34)$$

First we consider Calder's choice for b. Figure 5.2 shows the three stability regions $\mathcal{D}_{\mathrm{FD}_{\mathrm{Calder}}}, \mathcal{D}_{\mathrm{BD}_{\mathrm{Calder}}}, \mathcal{D}_{\mathrm{CD}_{\mathrm{Calder}}}$ resulting from this choice. The regions were produced by computing the stability conditions numerically across values of the complex λh^2-space. It is evident that the backward difference momentum descent method covers a larger spectrum, but for small $\mathrm{Re}(\lambda)$, no scheme allows non-zero imaginary part, but the central difference momentum descent may be more preferable in that case as it accepts a reasonable range of values along the real axis, but it also accepts values of λh^2 with smaller ratio $\mathrm{Re}(\lambda h^2)/|\mathrm{Im}(\lambda h^2)|$.

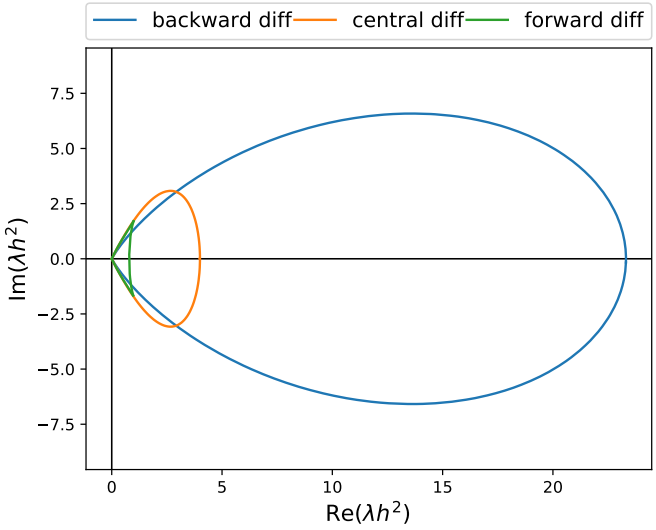


Figure 5.2: Stability regions $\mathcal{D}$ for the forward difference, backward difference and central difference momentum descent schemes when Calder's choice for b is used.

Next, we study the stability regions of the forwards, backwards and central differences schemes described in (5.32), (5.33) and (5.34), when the improved choice for b is used as in (5.25). Figure 5.3 shows an overlap of the three stability regions $\mathcal{D}_{\mathrm{FD}_{\mathrm{improved}}}, \mathcal{D}_{\mathrm{BD}_{\mathrm{improved}}}, \mathcal{D}_{\mathrm{CD}_{\mathrm{improved}}}$ resulting from the improved choice for b. We observe that similarly to the stability regions of Fig. 5.2 corresponding to Calder's choice, the backward difference momentum descent scheme covers a greater region of the real axis $\mathrm{Re}(\lambda h^2)$. However we see a significant improvement when $\mathrm{Re}(\lambda)$ is small. In fact, there is an obvious choice, the central difference momentum descent scheme, which allows very large $\mathrm{Im}(\lambda)$ to be treated. The forward difference momentum descent scheme is mostly unstable.

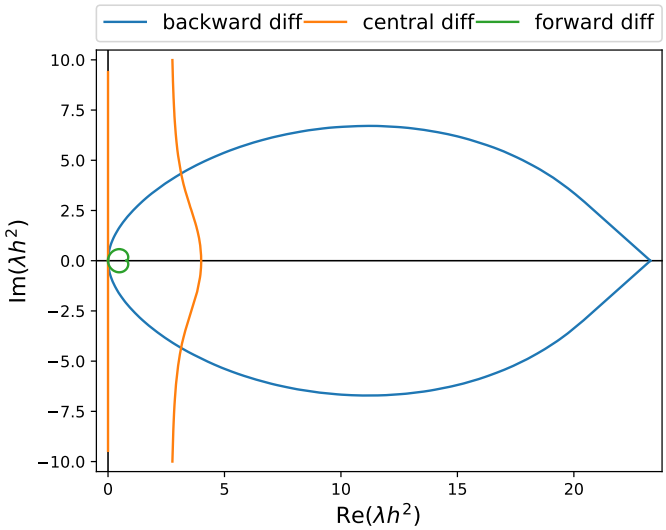


Figure 5.3: The stability regions in the complex plane for the forward difference, backward difference and central difference momentum descent schemes for the improved choice for b.

It is also interesting to inspect for each discretisation scheme the overlap of the stability regions resulting from the two choices of b. These comparisons are shown in Figs. 5.4, 5.5 and 5.6.

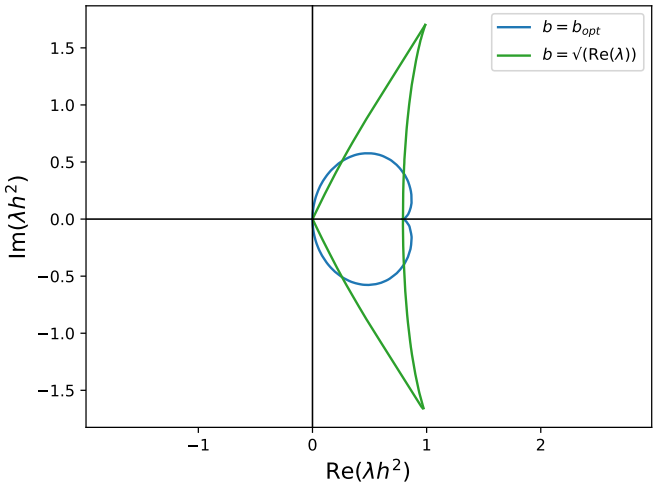


Figure 5.4: Stability regions of the forward difference momentum descent scheme for the Calder and improved choices of b, labeled $b = \sqrt{\mathrm{Re}(\lambda)}$ and b_{opt} respectively.

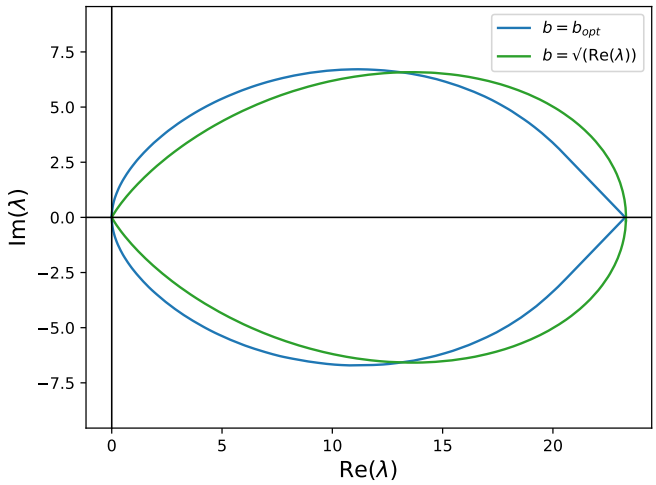


Figure 5.5: Stability regions for the backward difference momentum descent scheme for the Calder and improved choices of b, labeled $b = \sqrt{\mathrm{Re}(\lambda)}$ and b_{opt} respectively.

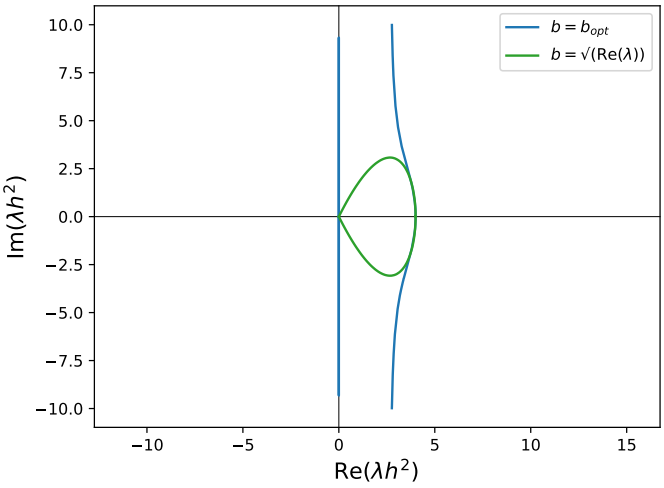


Figure 5.6: Stability regions for the central difference momentum descent scheme for the Calder and improved choices of b, labeled $b = \sqrt{\mathrm{Re}(\lambda)}$ and b_{opt} respectively.

It is apparent that, apart from the case of forward difference momentum descent, the improved choice b_{opt} in general improves the regions of stabilities. In fact for the central difference momentum descent, our new choice, permits large (in absolute value) imaginary parts, even for the case of small real parts. However one should note that a small $\mathrm{Re}(\lambda h^2)$, restricts the step size h thus the central difference is only a good choice in the case where $\mathrm{Im}(\lambda)$ is large.

5.4 A note on preconditioning

One may precondition the momentum descent schemes by simply preconditioning the forcing term $\mathbf{F}$ alone. For the string and NEB methods we assume the preconditioning equations of chapter 4. Notice that in this case, one expects that the eigenvalues of the preconditioned Jacobian will have $\mathrm{Re}(\lambda) \approx 1$ and $|\mathrm{Im}(\lambda)| < 1$. Recall from Sec. 5.2, that as $\mathrm{Im}(\lambda) \to 0$, $\beta_{\mathrm{opt}} = \mathcal{O}[\mathrm{Im}(\lambda)^{2/3}]$, then from (5.25), for a preconditioned Jacobian,

$$b_{\mathrm{opt}} = 2 + \mathcal{O}[|\mathrm{Im}(\lambda)|^{1/3}]. \tag{5.35}$$

We conclude that for the preconditioned case we expect $b_{\text{opt}} \approx 2$ to be a good choice.

5.5 Momentum descent for saddle point search methods

To apply our findings to saddle finding methods we look at the Jacobian matrices associated with each forcing term $\mathbf{F}$ for the string, NEB and dimer methods to get a sense of the eigenspectrum associated with each of them and determine which discretisation scheme should be used. We emphasise that the calculation of the Jacobian is not needed when using the momentum descent method. We will assume that we are in the vicinity of the equilibrium so that only a few eigenvalues of the Jacobian matrix may have non-positive real part and these are small in magnitude. In all cases, we consider the general preconditioned relations of chapter 4.

5.5.1 The Jacobian of the string and NEB methods

We investigate the Jacobian matrix $\mathbf{J}$ associated with the evolution directions $\boldsymbol{f}_n^k$ of each of the string and NEB methods. We once more assume a discretised path $\mathbf{X} \in \mathbb{R}^{N\times M}$

$$\mathbf{X} := \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix},$$

associated with images $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and let $\mathbf{F} \in \mathbb{R}^N \times \mathbb{R}^M$,

$$\mathbf{F} := \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_N \end{bmatrix},$$

be the direction of evolution of each of the images, associated with the string or the NEB method. The Jacobian of $\mathbf{F}$ is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{x}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_N}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{f}_N}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial \boldsymbol{f}_N}{\partial \boldsymbol{x}_N} \end{bmatrix}.$$

For the string method, the block matrices $\partial \boldsymbol{f}_n^{\mathrm{str}} / \partial \boldsymbol{x}_m$ associated with

$$\boldsymbol{f}_n^{\mathrm{str}} = -([\mathbf{P}_n^k]^{-1} - \boldsymbol{t}_{\mathrm{P},n} \otimes \boldsymbol{t}_{\mathrm{P},n}) \nabla_x V(\boldsymbol{x}_n) \tag{5.36}$$

of equation (4.10) are given by

$$\begin{aligned} \frac{\partial \mathbf{f}_n^{\mathrm{str}}}{\partial \boldsymbol{x}_{n-1}} &= \ell_n^{-1} \{ (\mathbf{t}_{\mathrm{P},n} \cdot \nabla V)(\mathbf{I} - \mathbf{t}_{\mathrm{P},n} \otimes \mathbf{P}_n \mathbf{t}_{\mathrm{P},n}) + \mathbf{t}_{\mathrm{P},n} \otimes [(\mathbf{I} - \mathbf{P}_n \mathbf{t}_{\mathrm{P},n} \otimes \mathbf{t}_{\mathrm{P},n}) \nabla V] \}, \\ \frac{\partial \mathbf{f}_n^{\mathrm{str}}}{\partial \boldsymbol{x}_n} &= (\mathbf{P}_n^{-1} - \mathbf{t}_{\mathrm{P},n} \otimes \mathbf{t}_{\mathrm{P},n}) \nabla \nabla V, \\ \frac{\partial \mathbf{f}_n^{\mathrm{str}}}{\partial \boldsymbol{x}_{n+1}} &= -\ell_n^{-1} \{ (\mathbf{t}_{\mathrm{P},n} \cdot \nabla V)(\mathbf{I} - \mathbf{t}_{\mathrm{P},n} \otimes \mathbf{P}_n \mathbf{t}_{\mathrm{P},n}) + \mathbf{t}_{\mathrm{P},n} \otimes [(\mathbf{I} - \mathbf{P}_n \mathbf{t}_{\mathrm{P},n} \otimes \mathbf{t}_{\mathrm{P},n}) \nabla V] \}, \\ \frac{\partial \mathbf{f}_n^{\mathrm{str}}}{\partial \boldsymbol{x}_m} &= \mathbf{O} \quad \text{if } m \notin \{n-1, n, n+1\}, \end{aligned} \tag{5.37}$$

where

$$\ell_n = \|\boldsymbol{x}_{n+1} - \boldsymbol{x}_{n-1}\|_{\mathbf{P}_n}, \tag{5.38}$$

and central differences where assumed for the evaluation of the tangents $\boldsymbol{t}_{\mathrm{P},n}$. Similarly, for the Jacobian of the NEB method, the respective block matrices $\partial \boldsymbol{f}_n^{\mathrm{neb}} / \partial \boldsymbol{x}_m$ associated with

$$\boldsymbol{f}_n^{\mathrm{neb}} = -(\mathbf{P}_n^{-1} - \boldsymbol{t}_{\mathrm{P},n} \otimes \boldsymbol{t}_{\mathrm{P},n}) \nabla_x V(\boldsymbol{x}_n) + \kappa \left(\boldsymbol{x}_n'' \cdot \mathbf{P}_n \boldsymbol{t}_{\mathrm{P},n} \right) \boldsymbol{t}_{\mathrm{P},n} \tag{5.39}$$

of equation (4.16), are given by

$$
\begin{aligned}
\frac{\partial \boldsymbol{f}_n^{\text{neb}}}{\partial \boldsymbol{x}_{n-1}} &= \frac{\partial \boldsymbol{f}_n^{\text{str}}}{\partial \boldsymbol{x}_{n-1}} && + \kappa \boldsymbol{t}_{\text{P},n} \otimes (\mathbf{P}_n \boldsymbol{t}_{\text{P},n}) \\
& && - \ell_n^{-1} \kappa \boldsymbol{t}_n \otimes \Big((\boldsymbol{x}_{n-1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n+1}) \cdot \mathbf{P}_n (\mathbf{I} - \boldsymbol{t}_n \otimes (\mathbf{P}_n \boldsymbol{t}_n)) \Big) \\
& && - \ell_n^{-1} \kappa (\boldsymbol{x}_{n-1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n+1}) \cdot \mathbf{P}_n \boldsymbol{t}_n \big(\mathbf{I} - \boldsymbol{t}_n \otimes (\mathbf{P}_n \boldsymbol{t}_n) \big), \\
\frac{\partial \mathbf{f}_n^{\text{neb}}}{\partial \boldsymbol{x}_n} &= \frac{\partial \mathbf{f}_n^{\text{str}}}{\partial \boldsymbol{x}_n} && - 2\kappa \boldsymbol{t}_{\text{P},n} \otimes \big(\mathbf{P}_n \boldsymbol{t}_{\text{P},n} \big), \qquad (5.40) \\
\frac{\partial \boldsymbol{f}_n^{\text{neb}}}{\partial \boldsymbol{x}_{n+1}} &= \frac{\partial \boldsymbol{f}_n^{\text{str}}}{\partial \boldsymbol{x}_{n+1}} && + \kappa \boldsymbol{t}_{\text{P},n} \otimes (\mathbf{P}_n \boldsymbol{t}_{\text{P},n}) \\
& && + \ell_n^{-1} \kappa \boldsymbol{t}_n \otimes \Big((\boldsymbol{x}_{n-1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n+1}) \cdot \mathbf{P}_n (\mathbf{I} - \boldsymbol{t}_n \otimes (\mathbf{P}_n \boldsymbol{t}_n)) \Big) \\
& && + \ell_n^{-1} \kappa (\boldsymbol{x}_{n-1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n+1}) \cdot \mathbf{P}_n \boldsymbol{t}_n \big(\mathbf{I} - \boldsymbol{t}_n \otimes (\mathbf{P}_n \boldsymbol{t}_n) \big), \\
\frac{\partial \boldsymbol{f}_n^{\text{neb}}}{\partial \boldsymbol{x}_m} &= && \mathbf{O} \text{ if } m \notin \{n-1, n, n+1\},
\end{aligned}
$$

In fact let us construct a toy model simulating the behaviour of this Jacobian. This will give us an idea of the eigen-spectrum that we expect to observe. Suppose we have an N-image path, where N is an odd natural number. Define the $N \times N$ matrix

$$
\mathbf{J} = \begin{bmatrix}
1 & 0 & & \dots & 0 \\
\frac{N}{2}\big(1 - 2\frac{1}{N-1}\big) & 1 & -\frac{N}{2}\big(1 - 2\frac{1}{N-1}\big) & \dots & \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
 & \dots & -\frac{N}{2}\big(1 - 2\frac{N-2}{N-1}\big) & 1 & \frac{N}{2}\big(1 - 2\frac{N-2}{N-1}\big) \\
0 & \dots & & 0 & 1.
\end{bmatrix} \qquad (5.41)
$$

which is such that the off-diagonal transport terms, are scaled by the inverse of the separation of images, $(1/N)^{-1}$ and vanish at the fixed points (the saddle point was assumed to be the central image). Now, consider the case $\mathbf{F} : \boldsymbol{x} \mapsto \mathbf{J}\boldsymbol{x}$, so that $\mathbf{J}$ coincides with the Jacobian of $\mathbf{F}$. The eigenvalues of $\mathbf{J}$ are such that $\text{Re}(\lambda_n) = 1$. From Eq. (5.14), this implies that at each b,

$$
g^+(b, \lambda_n) > g^+(b, \lambda_m) \iff \|\text{Im}(\lambda_n)\| < \|\text{Im}(\lambda_m)\|, \qquad (5.42)
$$

and therefore the $g^+(b)$ curves do not intersect. Take for example the case

$N = 7$. Figure 5.7 shows the $g^+(b)$ curves of the respective eigen-spectrum. As it was previously mentioned in Sec. 5.4, all eigenvalues, whose values are explicitly stated on the legend of Fig. 5.7, lie along $\mathrm{Re}(\lambda) = 1$ but further, notice that there are eigenvalues whose imaginary part is of the same order in magnitude, as their real part. This is an indication that the central differences might be more suitable for this system.

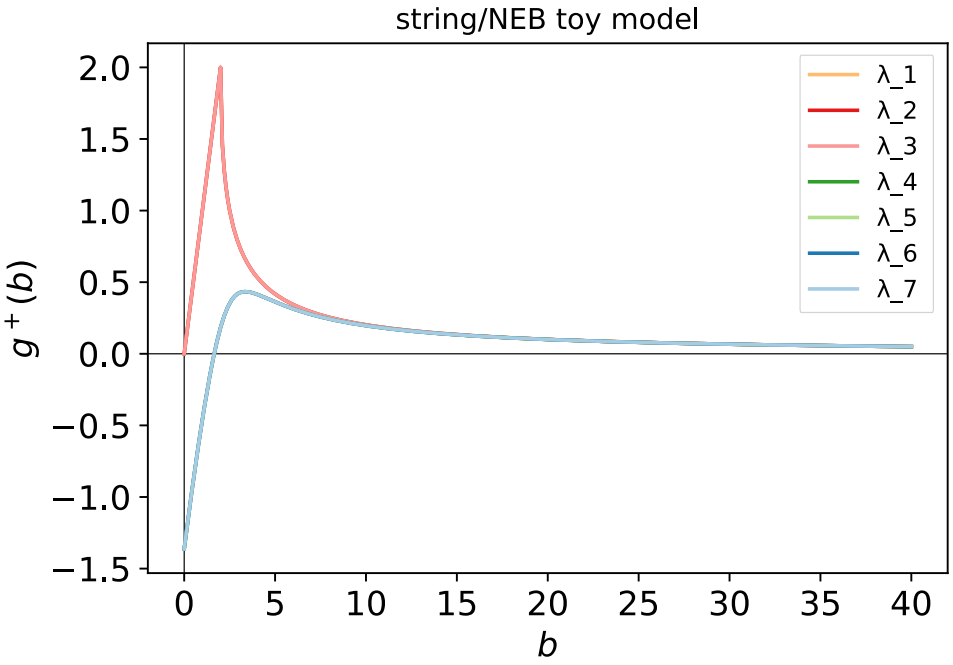


Figure 5.7: The $g^+(b)$ curves for the eigen-values of the Jacobian of the string-NEB type toy model given in (5.41). The pseudo path associated with this toy model consists of $N = 7$ images. The eigenvalues of the are Jacobian are: $\lambda_1 = \lambda_2 = \lambda_3 = 1$, $\lambda_4 = \lambda_6 = 1 - 1.65i$ (to 3 significant figures) and $\lambda_5 = \lambda_7 = 1 + 1.65i$ (to 3 significant figures).

From (5.42), and from Fig. 5.7, we see that finding the optimal choice for the momentum coefficient, b, for this toy model is trivial, as it suffices to minimise g^+ across all maxima $b = |\mathrm{Im}(\lambda_n)|/\sqrt{\mathrm{Re}(\lambda_n)}$, equivalently finding the λ_n which has the largest imaginary part, say λ_{maxIm} (degenerate eigenvalues are permitted) and then identifying the optimal value of $b_{\mathrm{opt}}(\lambda_{\mathrm{maxIm}})$ according to (5.25).

For N an integer taking values in the interval $[7, 34]$, we test the three discrete schemes for momentum descent as defined in Sec. 5.3.1, against the

simple steepest descent scheme. Figure 5.8 shows how the number of iterations needed for the residual error to become smaller than 10^{-2}, increases with the condition number of $\mathbf{J}$. We included steepest descent with adaptively chosen time step, for reference. We instantly identify that central differences is the correct choice offering the scaling we would expect for a momentum descent method, i.e. $\sim \kappa$, improving on the steepest descent where scaling is $\sim \kappa^2$.

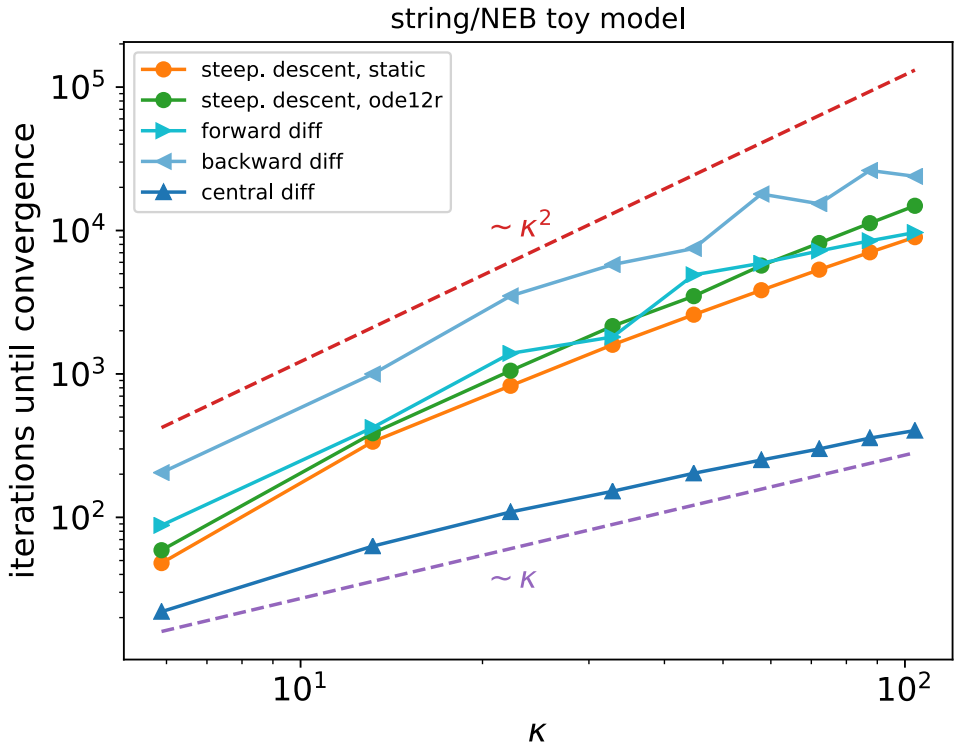


Figure 5.8: Number of iterations vs condition number for the steepest-descent and momentum-descent methods. The string-type toy model (5.41) was used. Reference lines show scaling of the number of iterations proportional to κ^2 and κ.

Fixing the number of images to $N = 19$, we investigate how the residual error of this toy model scales as a function of the number of iterations, for the three discrete schemes of Sec. 5.3.1. Each iteration of the methods is considered to carry computational cost equivalent to a unit. The results, shown in Fig. 5.9, are compared with the simple steepest descent method, with static step. These confirm the earlier prediction from our analysis and observation that the central difference momentum descent scheme offers the

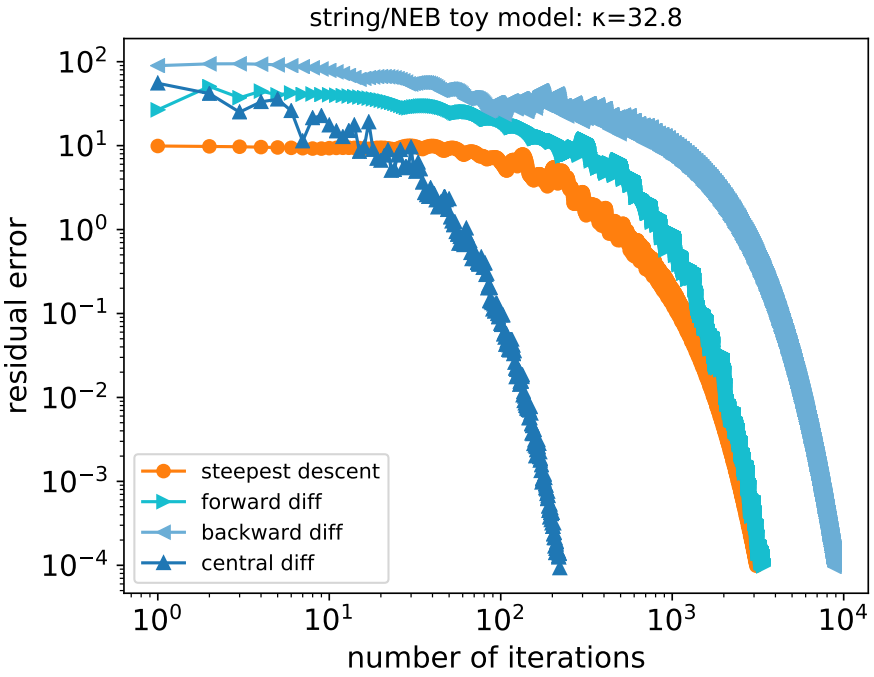


Figure 5.9: Residual error against number of iterations for the string/NEB toy model (5.41).

greatest speedup. Consequently whenever we refer to momentum descent in string/NEB type methods, we will assume the use of the central difference scheme.

5.5.2 The Jacobian of the dimer method

We follow the same process to investigate the Jacobian matrix $\mathbf{J}^{\text{dimer}}$ of the dimer method. Take $\mathbf{X} \in \mathbb{R}^{2\times M}$

$$\mathbf{Z} := \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{bmatrix},$$

associated with dimer pair $(\boldsymbol{x}, \boldsymbol{v})$ and let $\mathbf{F} \in \mathbb{R}^{2\times M}$,

$$\mathbf{F} := \begin{bmatrix} \boldsymbol{f}_{\text{x}} \\ \boldsymbol{f}_{\text{v}} \end{bmatrix},$$

be the driving force associated with the dimer method coming from expressions (2.80) and (2.81). The Jacobian of $\mathbf{F}$ is given by

$$\mathbf{J}^{\text{dimer}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \tilde{\boldsymbol{v}}} \\ \frac{\partial \boldsymbol{f}_{\text{v}}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{f}_{\text{v}}}{\partial \tilde{\boldsymbol{v}}} \end{bmatrix},$$

where

$$\boldsymbol{f}_{\text{x}} = -(\mathbf{P}(\boldsymbol{x})^{-1} - 2\tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}})\nabla V(\boldsymbol{x}) \tag{5.43}$$

$$\boldsymbol{f}_{\text{v}} = -(\mathbf{P}(\boldsymbol{x})^{-1} - \tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}})\frac{\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big) - \nabla V(\boldsymbol{x})}{L}. \tag{5.44}$$

The block matrices of equation (5.5.2) are given by

$$\begin{aligned}
\frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \boldsymbol{x}} &= -\big(\mathbf{P}^{-1} - 2\tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}}\big)\nabla\nabla V(x) \\
\frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \tilde{\boldsymbol{v}}} &= 2\big(\tilde{\boldsymbol{v}} \cdot \nabla V(x)\big)\Big(I - \tilde{\boldsymbol{v}} \otimes \big(\mathbf{P}\tilde{\boldsymbol{v}}\big)\Big) + 2\tilde{\boldsymbol{v}} \otimes \Big[\Big(I - \big(\mathbf{P}\tilde{\boldsymbol{v}}\big) \otimes \tilde{\boldsymbol{v}}\Big)\nabla V(x)\Big] \\
\frac{\partial \boldsymbol{f}_{\text{v}}}{\partial \boldsymbol{x}} &= -L^{-1}\Big(\mathbf{P}^{-1} - \tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}}\Big)\Big(\nabla\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big) - \nabla\nabla V(x)\Big) \\
\frac{\partial \boldsymbol{f}_{\text{v}}}{\partial \tilde{\boldsymbol{v}}} &= -\big(\mathbf{P}^{-1} - \tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}}\big)\nabla\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big) \\
&\quad + \big[\big(\mathbf{P}^{-1} - \tilde{\boldsymbol{v}} \otimes \tilde{\boldsymbol{v}}\big)\nabla\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big)\tilde{\boldsymbol{v}}\big] \otimes \big(\mathbf{P}\tilde{\boldsymbol{v}}\big) \\
&\quad + L^{-1}\big(\tilde{\boldsymbol{v}} \cdot \big(\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big) - \nabla V(x)\big)\big)\big(I - \tilde{\boldsymbol{v}} \otimes \big(\mathbf{P}\tilde{\boldsymbol{v}}\big)\big) \\
&\quad + L^{-1}\tilde{\boldsymbol{v}} \otimes \big[\big(I - \big(\mathbf{P}\tilde{\boldsymbol{v}}\big) \otimes \tilde{\boldsymbol{v}}\big)\big(\nabla V\big(\boldsymbol{x} + L\tilde{\boldsymbol{v}}\big) - \nabla V(x)\big)\big]
\end{aligned}$$

The study of the eigenspectrum of $\mathbf{J}^{\text{dimer}}$ was performed numerically in the results Sec. 5.6 below, but because at the saddle $\frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \tilde{\boldsymbol{v}}} = 0$ and the $\frac{\partial \boldsymbol{f}_{\text{x}}}{\partial \boldsymbol{x}}$, $\frac{\partial \boldsymbol{f}_{\text{v}}}{\partial \tilde{\boldsymbol{v}}}$ are symmetric, the eigenspectrum near the saddle is real [2] and thus close to the saddle we expect that the eigenspectrum will have small imaginary parts. From the stability regions of Fig. 5.3, the backwards momentum descent method would be the most appropriate choice.

[2] Let $\mathbf{A}, \mathbf{B}$ be real and symmetric square matrices and $\mathbf{V}\mathbf{A}\mathbf{V}^{\text{T}} = \Lambda_{\text{A}}$ and $\mathbf{U}\mathbf{B}\mathbf{U}^{\text{T}} = \Lambda_{\text{B}}$ the corresponding eigendecompositions, with real eigenvalues . Let $\mathbf{C}$ be a square matrix, then the following is true $\begin{pmatrix} \mathbf{V} & 0 \\ 0 & \mathbf{U} \end{pmatrix}\begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{C} & \mathbf{B} \end{pmatrix}\begin{pmatrix} \mathbf{V}^{\text{T}} & 0 \\ 0 & \mathbf{U}^{\text{T}} \end{pmatrix} = \begin{pmatrix} \mathbf{\Lambda}_{\text{A}} & 0 \\ 0 & \mathbf{\Lambda}_{\text{B}} \end{pmatrix}$ and the eigenvalues of $\begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{C} & \mathbf{B} \end{pmatrix}$ are also real.

5.6 Results

We look once more at the vacancy and screw dislocation systems we have analysed in Sec. 2.3.1 and Sec. 2.3.2. For each system, we confirm the convergence speed up for momentum descent for the optimisation problem, and we demonstrate its effectiveness for the string, NEB and dimer methods. For the specific case of the vacancy in a 2D triangular lattice modelled by the Lennard-Jones potential, we also look at the Jacobian to confirm the choice of discretisation scheme for the optimisation, string and NEB methods and to identify the appropriate discretisation scheme for the dimer method. The values of h and b were determined numerically, by comparing the convergence speeds of repeated calculations for a range of values.

5.6.1 Vacancy in a 2D Lennard Jones lattice

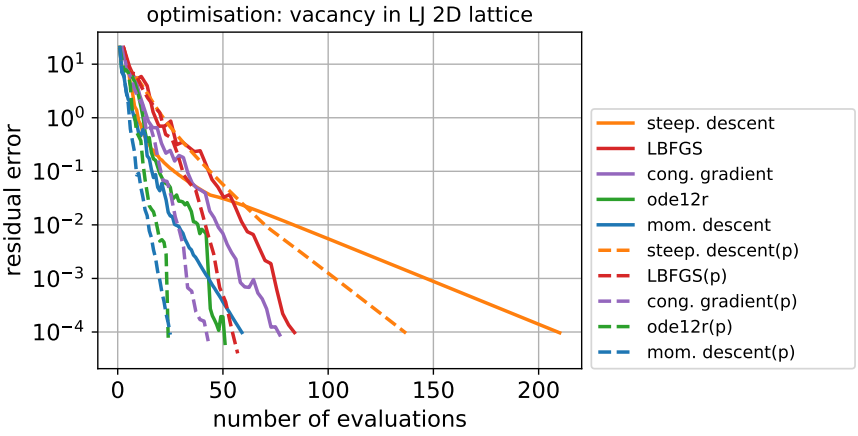


Figure 5.10: Convergence for minimisation of a system describing a vacancy in 60-atom 2D cell modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. We compare the momentum descent method against the simple steepest descent, LBFGS, conjugate gradient and steepest descent with adaptive time stepping (*ode12r*).

Consider a vacancy in a two dimensional 60-atom triangular lattice. A Lennard-Jones potential models the system (see Sec. 2.2.3), with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. The EXP preconditioner described in Sec 4.4.1 with parameters $A = 3.0$ and $r_{\text{cut}} = 2.5$ is used for preconditioning. The vacancy originally is located near the centre of the cell. We relax the configuration to

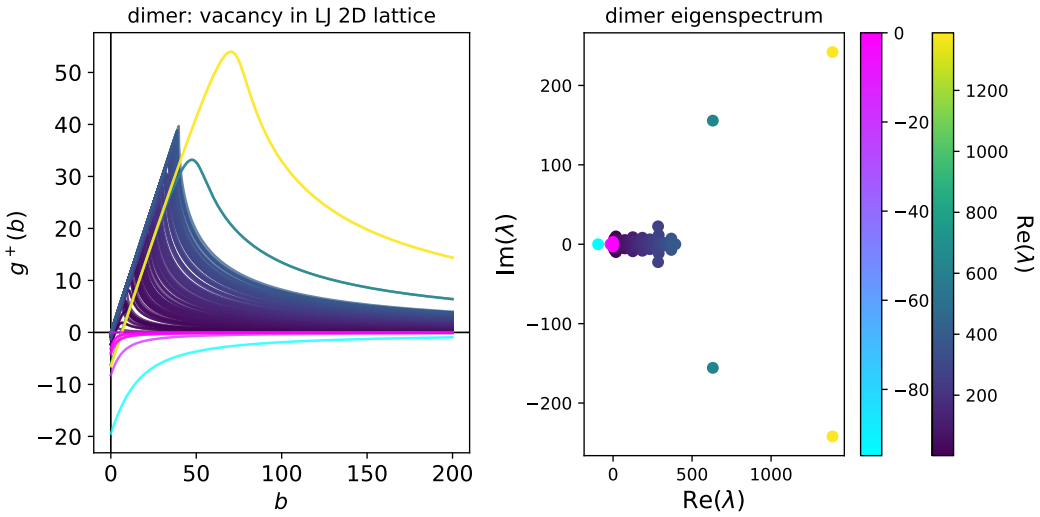


Figure 5.11: The left panel shows g^+ against b for the eigenspectrum, which is shown on the right panel, for the Jacobian of the dimer method, for a 60-atom 2D cell modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. Different colour maps have been used to separate eigenvalues of positive and negative real parts.

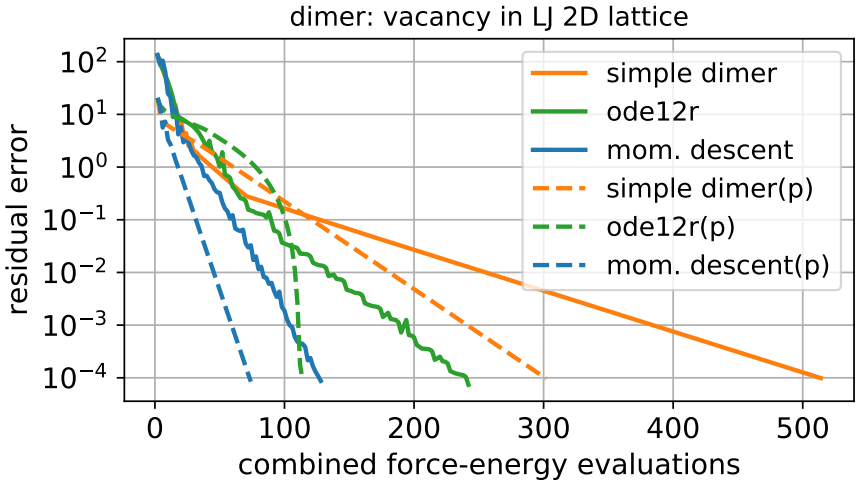


Figure 5.12: Convergence of the dimer method for a system describing a 60-atom 2D cell modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. The initial configuration is near the local minimum. We compare the convergence rates of the momentum descent method against the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

the nearest local energy minimum by means of the momentum descent method. Figure 5.10 shows the convergence rate in comparison to the simple steepest

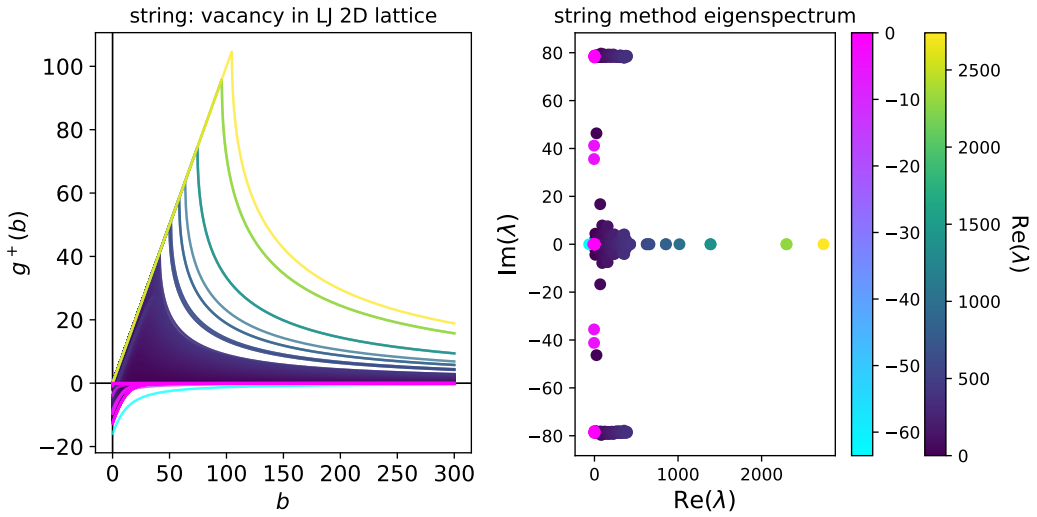


Figure 5.13: The left panel shows g^+ against b for the eigenspectrum of the Jacobian of the string method. On the right panel the eigenspectrum is depicted. The system describes a 60-atom 2D cell modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. Different colour maps have been used to separate eigenvalues of positive and negative real parts. This refers to a 9-image path whose end points are free to relax to the local minima.

descent, steepest descent with *ode12r* adaptive time stepping, LBFGS and conjugate gradient algorithms. We remind the reader that the aim of this section is not to outperform LBFGS or conjugate gradient, but to create a scheme that is transferable to saddle point search methods and that the comparison here is just for completeness of the results. The preconditioned variants, where available, are also pictured.

From Fig 5.10, the momentum descent method is as effective as the adaptive steepest descent method and in this case it is more efficient than the other adaptive schemes. The effective speed up is measured against the simple descent method where a static step is also used and this speedup is of a factor of 4 for the un-preconditioned case, and of a factor of 5 for the preconditioned case. Focusing on the un-preconditioned momentum descent case, one may distinguish clearly the regions of pre-asymptotic and asymptotic convergence where in the latter case an adaptively chosen step is expected to offer an even greater speedup. The same may be observed in the preconditioned case but the difference is not that apparent to the naked eye.

Starting from a configuration near the midpoint of the minimum and the

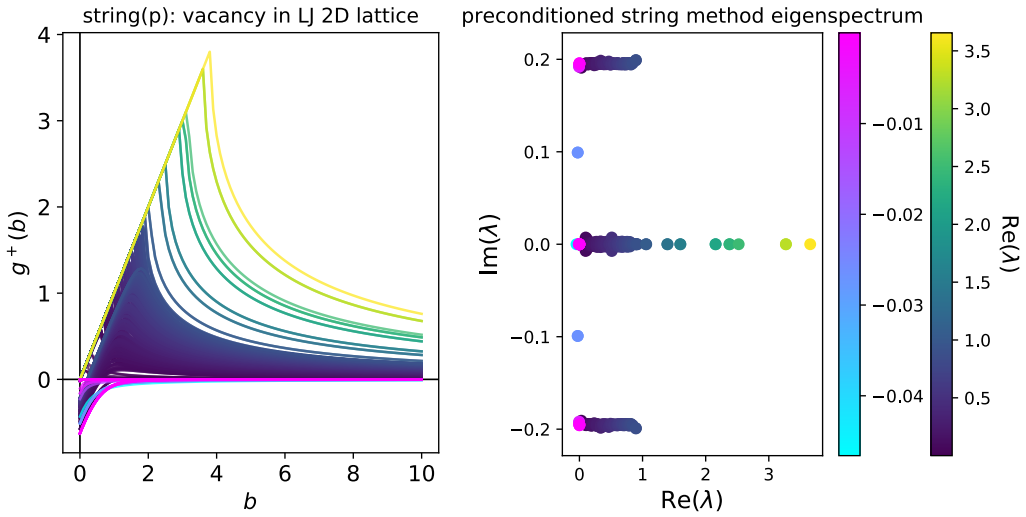


Figure 5.14: On the left panel we have the values of $g^+(b)$ for the eigenspectrum of the Jacobian of the preconditioned string method, for a 60-atom 2D cell modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. The eigenspectrum is shown on the right panel and different colour maps have been used to separate eigenvalues of positive and negative real parts. This refers to a 9-image path whose end points are free to relax to the local minima.

saddle, we use the dimer method to find the saddle for the vacancy migrating by one lattice spacing. Figure 5.11 shows on the left the $g^+(b)$ curves for the eigenspectrum of the Jacobian of the dimer method, together with a scatter plot of the eigenspectrum in the complex plane (shown on the right). The scatter plot suggests that using backward differences is a good choice. Further, all eigenvalues with negative real part have $\mathrm{Im}(\lambda) = 0$. The $g^+(b)$ curves on the left hand side plot, suggest that the contribution of eigenvalues with negative real part is not insignificant and thus we expect that the momentum descent method will converge slowly without preconditioning.

In Fig. 5.12, the convergence rate for the dimer method is depicted, comparing the momentum descent method against the steepest descent method with static and *ode12r* adaptive time step selection, with and without preconditioning. In this case the momentum descent method offers an even better performance than the adaptive step selection (almost a 2-fold speed-up for both the unpreconditioned and preconditioned cases), whereas the effective speed-up against the static method is more than a factor of 3. This suggests

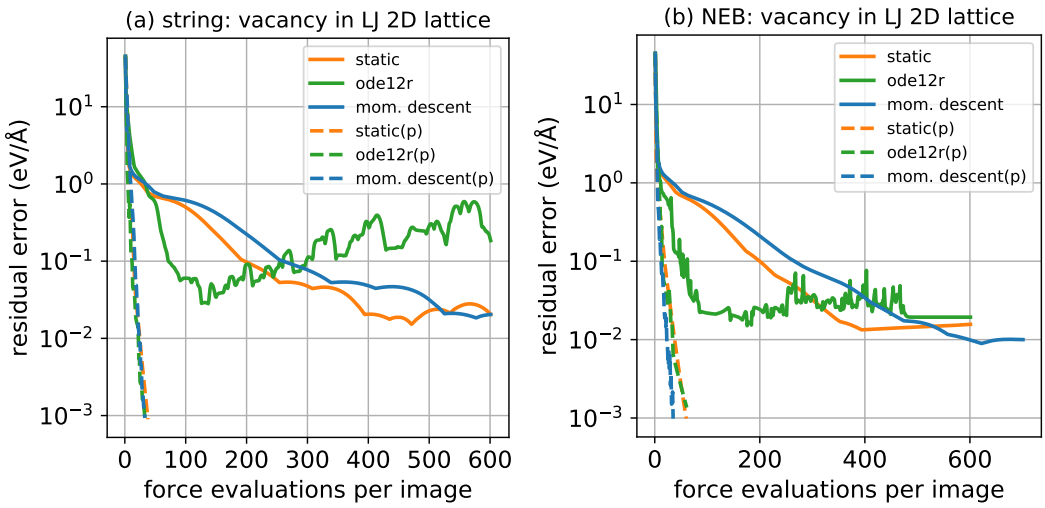


Figure 5.15: Convergence rate of the (a) string and (b) NEB methods for a vacancy in a 60-atom 2D triangular lattice modelled with the Lennard-Jones potential, with parameters $\epsilon = 1.0$, $\sigma = 2^{-\frac{1}{6}}$. A 9 image path was used with the end points free to relax to the respective local minima. We compare the convergence rates of the momentum descent method against the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

that an adaptive step for the momentum descent method might offer an even greater speed up.

Finally, we look at the string and NEB methods for identifying the MEP of the same vacancy migration. We use a 9-image path, with the end points free to move to the respective local minima. A linear interpolation of the two end points was used to initialise the path. Figures 5.13 and 5.14 illustrate the eigenspecrum of the Jacobian of the unpreconditioned and preconditioned string method on the right hand side, and the respective $g^+(b)$ curves on the left. This reinforces our earlier observation of Sec. 5.5.1 that central differences should be used for the momentum descent scheme. Further, comparison of the two right hand side scatter plots demonstrates the improvement on the associated condition number from using the preconditioner, as the majority of the eigenvalues of the preconditioned string are now clustered in regions of $\mathrm{Re}(\lambda) = 1$ with the exception of a few outliers.

One can see from the plots that choosing b close to half the improved choice b_{opt} corresponding to the eigenvalue of greatest magnitude will be sufficient to converge the majority of the eigenmodes fast. For the preconditioned

case, this is in fact close to 2 as expected. The respective plots for the NEB method are omitted as they offer similar behaviour.

The convergence plots for the string and NEB methods are shown in Figure 5.15. The momentum descent method applied to the unpreconditioned string and NEB methods failed to converge, similarly to the unpreconditioned steepest descent method. In fact it offered similar convergence rates to the static steepest descent method. The preconditioned case on the other hand offered similar rapid speedup, comparable to the preconditioned steepest descent with static and adaptive time step.

5.6.2 Vacancy in a Cu supercell

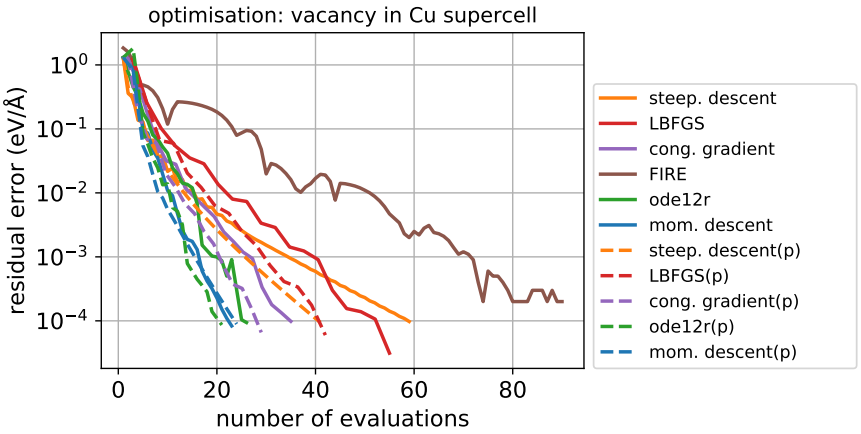


Figure 5.16: The system describes a vacancy confined in a 107-atom Cu fcc supercell modelled with the Morse potential, with parameters $A = 4.0$, $\epsilon = 1.0$ and nearest boundary distance $r_0 = 2.55$ Å. Periodic boundary conditions were applied. Beginning from a configuration near the local minimum, we relax the system using the momentum descent method. We compare against the simple steepest descent, LBFGS, conjugate gradient, FIRE and steepest descent with adaptive time stepping (*ode12r*).

The same tests were carried out for a vacancy in a fixed cell and periodic boundary conditions for a 107-atom Cu FCC supercell modelled with the Morse potential with parameters $A = 4.0$, $\epsilon = 1.0$ and nearest boundary distance $r_0 = 2.55$ Å. For the preconditioned cases, configurations were preconditioned with the EXP preconditioner introduced in Packwood *et al.* [71] with parameters $A = 3.0$, $r_{\text{cut}} = 2.2$ and $r_0 = 5.62$ Å.

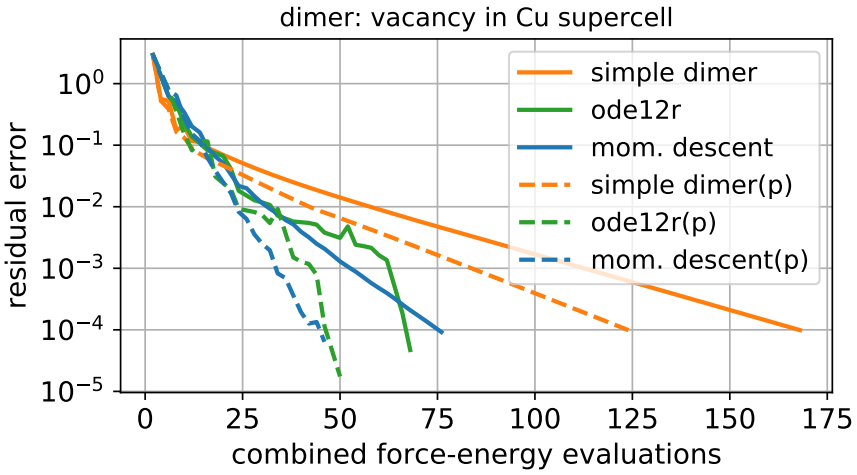


Figure 5.17: Convergence rate of the dimer method for a vacancy in a 107-atom Cu fcc supercell modelled with the Morse potential, having $A = 4.0$, $\epsilon = 1.0$ and nearest boundary distance $r_0 = 2.55$ Å. The system was initialised near the local minimum. The convergence rate of the momentum descent method was compared against that of the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

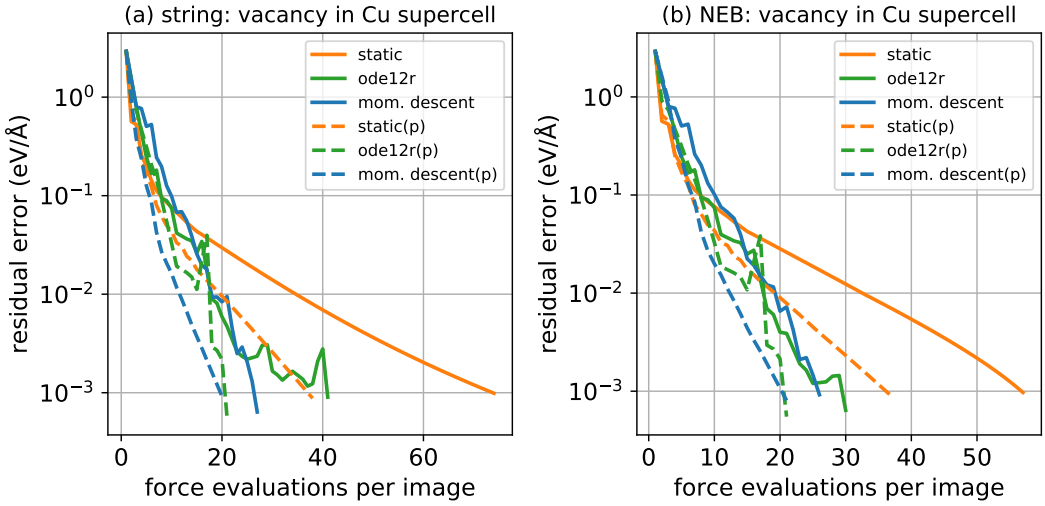


Figure 5.18: Convergence rate of the (a) string and (b) NEB methods for a vacancy in a 107-atom Cu fcc supercell with periodic boundary conditions, modelled with a Morse potential of patrameters $A = 4.0$, $\epsilon = 1.0$ and $r_0 = 2.55$ Å. A 5 image path was used and the end points are free to move to the respective local minima. The convergence rate of the momentum descent method is compared against that of the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

For the optimisation problem, we initialise our configuration near the local minimum. We compare the momentum descent method against steepest descent with static and adaptive stepping (*ode12r*), the LBFGS method, conjugate gradient and FIRE. The unpreconditioned and preconditioned momentum descent methods are comparable in performance and these are also comparable to the performance of the *ode12r* methods. The effective speedup relative to the static steepest descent is of the order of 3 for the unpreconditioned results and of the order of 2 for the preconditioned results.

To test the performance of the momentum descent dimer method, we initialise the system near the mid point between the minimum and the saddle. We compared the convergence rate of the method against the steepest descent dimer method with static and adaptive *ode12r* step selection. The results are depicted in Fig. 5.17. One observes a similar behaviour, as in our previous results. The acceleration of the static momentum descent method is comparable to the speedup gained from using an adaptive *ode12r* step in steepest descent, giving a net speedup greater than a factor of 2 for the unpreconditioned results and a factor of 2.5 for the preconditioned results.

Finally, testing the string and NEB methods, one may identify clearly on Figure 5.18, the point where the system enters the asymptotic regime and the momentum descent method could have benefitted significantly from the use of an adaptive step in the way the simple steepest descent benefits from the use of the *ode12r* adaptive step selection. The effective speedup from the static steepest descent is of the order of 3 for the the unpreconditioned string, and 2 for the unpreconditioned NEB method and the respective speedup factors for the preconditioned cases are of the order of 2. We do notice that the speedup is more significant in the unpreconditioned case and this is expected as in the preconditioned case the condition number is approaching 1.

5.6.3 Vacancy in a W supercell

We now look at a vacancy in a W BCC supercell which we model with the EAM4 potential and for which we impose periodic boundary conditions. For preconditioning a configuration in the system we use the FF preconditioner of Mones *et al.* [2]. We carry out the same set of tests.

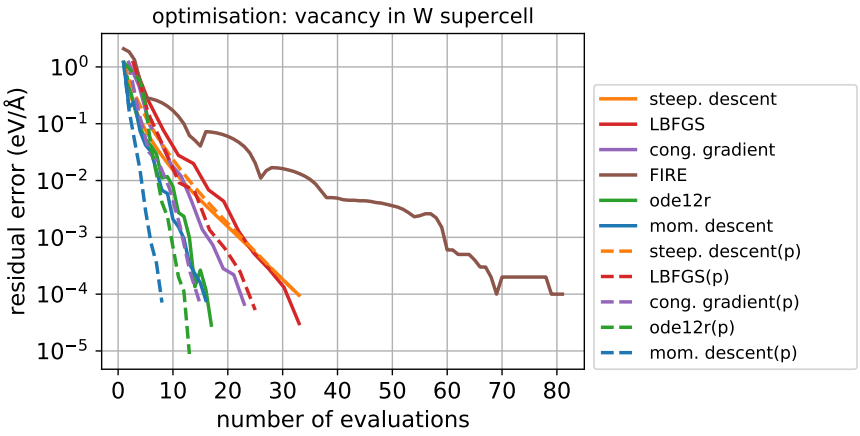


Figure 5.19: The system is a vacancy in a 53-atom W BCC supercell modelled with the EAM4 potential. Periodic boundary conditions were applied. Initialising our system at a configuration near the local minimum, we relax the system using the momentum descent method. The performance of the momentum descent method is compared against the performance of the simple steepest descent, LBFGS, conjugate gradient, FIRE and steepest descent with adaptive time stepping (*ode12r*). Preconditioning was also used if available.

First we initialise the system near the minimum. We want to look at the performance of the momentum descent method in geometry optimisation. We compare its convergence against the steepest descent method with static and adaptive step selection (*ode12r*), the LBFGS, conjugate gradient and FIRE methods. If available, the preconditioned variants were also used. Figure 5.19 shows the results. In this example we see that momentum descent with preconditioning outperforms all other schemes, although the performance of the preconditioned *ode12r* and conjugate gradient methods is comparable. The effective speedup in comparison to the static steepest descent method is of a factor of 2 for the unpreconditioned case and of a factor of 4 for the preconditioned case.

To apply the dimer method, we initialised the system as we did before by taking a point near the mid point of a local minimum (the vacancy being at the centre of the cell) and a nearby saddle point. As shown in Fig. 5.20, the momentum descent method offered an effective speed-up greater than a factor of 2 for both the unprconditioned and preconditioned cases, when compared to the static steepest descent. Once more the importance of an adaptively chosen step is highlighted.

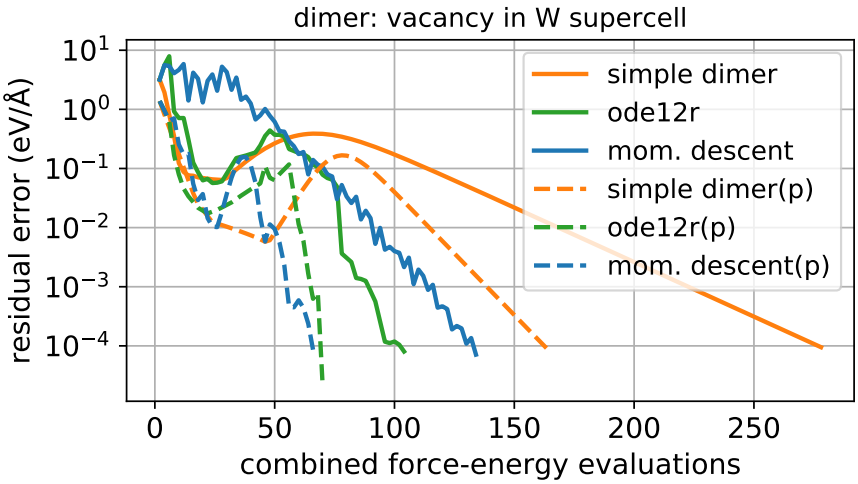


Figure 5.20: Convergence of the dimer method for a vacancy in a 53-atom W bcc supercell. The EAM4 potential was used to model the system and periodic boundary conditions were used. The system was initialised near the local minimum. The convergence speed of the momentum descent method was compared against that of the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

Finally, to test the string and NEB methods, we use a 5-image path which we initialise by linearly interpolating the initial and final states. We compare the results against the steepest descent with static and adaptive *ode12r* step selection. The results are shown in Fig. 5.21, where we observe that for the NEB method, the momentum descent method did not perform as expected in the preconditioned case. In fact the preconditioned steepest descent performs better than both the *ode12r* and momentum descent schemes in the case of the NEB method, but notice that all preconditioned cases offer the same convergence rate in the pre-asymptotic regime. For the momentum descent method it is apparent that a different step size should be used in the asymptotic regime and thus the method would benefit from an adaptively chosen step size, whereas for the *ode12r* it is the case of a suboptimal step size guess. For the string method the performance is similar to what we have seen in earlier tests. The momentum descent method performs as well as steepest descent with *ode12r* giving a 2-fold effective speedup for both the unpreconditioned and preconditioned cases.

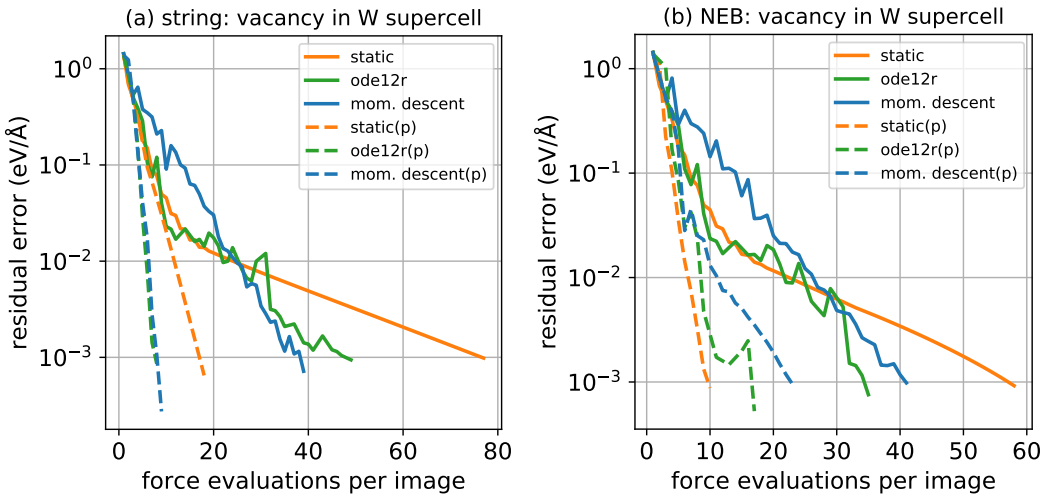


Figure 5.21: Convergence of the (a) string and (b) NEB methods for a vacancy in a 53-atom W bcc supercell, modelled with the EAM4 potential and periodicity applied on the boundaries. A 5 image path was used with the end points free to move to the respective local minima. We compare the convergence speeds of the momentum descent method to that of the simple steepest descent and steepest descent with adaptive time stepping (*ode12r*).

5.6.4 Screw Dislocation

For our final example, we study a $\frac{1}{2}\langle 111\rangle$ screw dislocation in a 1489-atom W BCC structure confined in a cylinder of radius equal to 40 Å and surrounded by a cylindrical shell of 565 clamped atoms. Periodic boundary conditions were applied along the dislocation line (z) direction. The system is simulated with the EAM4 potential.

Starting from a configuration near a local minimum, we relax the system to equilibrium and plot the results in Fig. 5.22a.

We then consider the transition of the dislocation advancing by one glide step. We first use the dimer method to identify the saddle of this transition. The results are shown in Fig. 5.22b.

Finally, we use the string and NEB methods to find the MEP of the transition. To do so, we use a 9-image path, for which we fixed the ends of the path at the local minima. The results are shown in Figs. 5.22c and 5.22d.

In all four cases, the unpreconditioned momentum descent method exhibits a significant speed up relative to the static steepest descent scheme, of a factor 3 for optimisation and of a factor 5 for all other methods, but further,

the method exhibits a speed up of the order of 1.5 for the optimisation, a speed up of order 3 for the dimer method and speed ups of factors 2.5 and 2 for the string and NEB method when compared to the adaptive *ode12r* scheme. For the preconditioned results, the 3 schemes are comparable, with the exception of the dimer method where a 2-fold speed up was observed for the momentum descent and *ode12r* schemes in comparison to the simple static dimer.

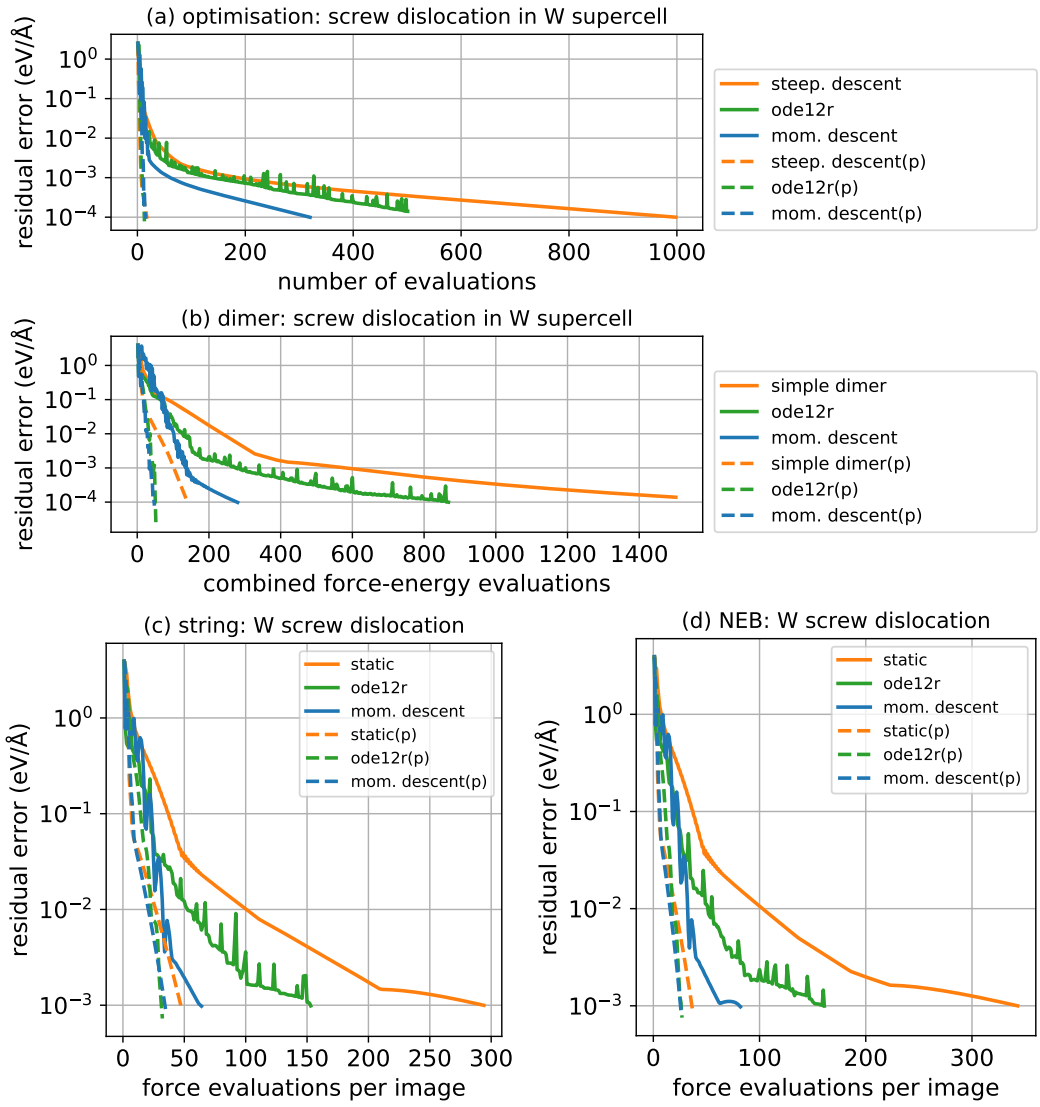


Figure 5.22: Convergence of the (a) optimisation, (b) dimer (c) string and (d) NEB methods for a screw dislocation in a 1489-atom W bcc structure confined in a cylinder of radius equal to R = 40Å and periodic boundary conditions along the dislocation line. Atoms outside outer radius of R were clamped.

5.6.5 Conclusion

Momentum descent, or accelerated gradient descent methods are 2-step discrete schemes that may be used to accelerate steepest descent (or gradient descent) optimisation schemes by improving the scaling of convergence with the condition number. In this chapter we demonstrated how the momentum descent method may be used to accelerate the saddle point search methods, the dimer method and the double ended string and NEB methods. We have seen that there is a unique optimal choice for the momentum coefficient b corresponding to each eigen-direction of the Jacobian of the forcing term of each method.

We investigated three discretisations of momentum descent based on forward, backward and central differences and we found the stability regions for each of them. This helped us identify a suitable scheme for each of the optimisation, dimer, string and NEB methods and gave us a heuristic way of identifying suitable values for the momentum coefficient.

Further it showed that $b \approx 2$ is a good choice and would be effective across all eigendirections of the Jacobian in the preconditioned case. Preconditioning of the momentum descent method was performed so that only the forcing term is preconditioned as proposed in Ch. 4. Our results suggested that the momentum descent method offers a significant speedup which is comparable to that of the adaptive *ode12r* method despite its simplicity. Further, our results suggested that an adaptive method for choosing the pair (b, h) of the momentum descent method can lead to further speedups, which we will explore in future work.

An open source prototype implementation of the techniques described in this chapter is available at `https://github.com/cortner/SaddleSearch.jl`.

Chapter 6

Conclusions

Computational simulations of the behaviour of physical systems can be challenging because of the presence of spontaneous rare transitions. Simulations running on the atomistic/molecular level are not suitable for identifying these rare events, due to the discrepancy of time scales. Instead, one should use simulations on the mesoscale [5] but to do so it is necessary that one knows the transition rate of the event under study.

As we have discussed in Chapter 2 this is possible if we first identify the energy barrier that the system needs to overcome for a transition to occur. In fact, the transition rate scales exponentially with the energy barrier [4, 6, 7], thus an accurate calculation of the latter is essential.

In this thesis we looked at the most widely used methods for identifying transition energy barriers, i.e. methods which identify the saddle point of the energy landscape separating the initial and final states. These are iterative schemes, grouped in two families: (a) methods that evolve a chain of images, connecting the reactive and product states of the transition, until convergence to the MEP of the transition and (b) 'walker' methods evolving a single state, together with an associated orientation and transition direction, until it converges to the saddle point. Of the first kind, we look at the NEB method [13, 14] which includes an elastic interaction between neighbouring images and the string method [10, 11, 12] which constrains the images of the chain to be at fixed relative distances at all times. Of the second family we investigate the dimer method [9].

It is commonly acknowledged that these methods often converge slowly

due to ill-conditioning of the potential of the system. Further, although adaptive schemes have been developed in the past to select the step length for the iterative step, these are not in general robust techniques, causing instabilities and kink formation along the evolved paths.

My goal has been the exploration of numerical optimisation techniques in conjunction with saddle point search methods in order to accelerate convergence and increase robustness.

In Chapter 3, we express the string and NEB methods as first order ODE problems whose steady state solution defines the MEP. Similarly, the dimer method is expressed as a first order ODE problem in terms of a fictitious time coordinate whose steady state solution is the saddle point of the transition. We were then able to use an appropriate adaptive numerical ODE solver to numerically integrate this ODE and in doing so we select appropriate step lengths for the iterative step. We choose the simple *ode12* method which uses a first and a second order approximation of the solution to estimate the local error. However, as we converge to an equilibrium of the ODE problem, as explained in [67, 68], the adaptive ODE solver underestimates the local error causing large time steps to be used incorrectly. This issue was observed in our results as well. We resolve this issue by introducing an intermediate line-search step which ensures that the accepted step at least satisfies a sufficient decrease condition on the residual. The resulting scheme, *ode12r*, is a robust adaptive selection method which often enjoys additionally a rapid convergence to equilibrium, due to the efficiency with which the step lengths are selected. We also saw that this method may be applied to optimisation directly, and the results compete with other fast adaptive techniques, such as LBFGS [54], despite its relative simplicity.

In simple optimisation, Newton's method offers the fastest convergence to an equilibrium and it is not sensitive to ill-conditioned potentials. However, it requires the evaluation of the Hessian of the potential which is not a feasible calculation for large systems and complex models, such as DFT. A familiar technique in numerical analysis is preconditioning. A preconditioner can be viewed as either a coordinate mapping in configuration space capturing the curvature of the potential, or as an approximation of the Hessian in analogy to Newton's scheme. In Chapter 4, we adopted the first view, so we constructed

a preconditioning scheme for string and NEB which utilises one geometry optimisation preconditioner for each image. Our proposed scheme uses force calculations only, and can therefore be used straightforwardly with hybrid models such as QM/MM which use force matching on the boundary between two modelling treatments. We demonstrated that for accurate preconditioner representations, our method is capable of achieving speed ups of factors as high as 6 over conventional approaches. The gain is expected to grow with system size. In the case of a DFT model for a small system size, where a closely fitted preconditioner was not available, we still observed a significant 2-fold speed up. These results are auspicious and suggest that further work should be conducted in order to construct higher accuracy preconditioners and achieve even higher convergence speeds. Additionally, we observed that preconditioning increased the robustness of these methods and improved accuracy.

Another observation is that the proposed preconditioning scheme does not treat ill-conditioning arising from longitudinal interactions (e.g. the elastic interactions of the NEB method), but only the effects of ill-conditioning in the normal direction. Work has already began in generalising the preconditioning scheme to include treatment of tangential effects. The idea is to construct a preconditioner for the force terms of the string, NEB or dimer methods, which captures the curvature of their associate Jacobian matrix.

Finally, in Chapter 5, we explore accelerated gradient descent methods [57, 58], also known as momentum descent methods which are primarily used in optimisation in place of the simple steepest descent method. These are two-step iterative methods, which use a momentum term to accelerate the convergence of the steepest descent method to scales better with condition number. I showed how the momentum descent method may be adapted to saddle point search methods and found the necessary conditions the momentum coefficient needs to satisfy. Three discretisation schemes were introduced, which use forward, backward and central differences to approximate the differential operators involved. We studied these discretisation schemes in terms of stability and based on these stability results we explain that the central differences momentum descent scheme is the most appropriate choice for the string and NEB methods, whereas the backwards differences is the most appropriate choice for the dimer method and for simple optimisation.

Preconditioning the momentum descent scheme was performed by means of preconditioning the forcing term associated with each of the string, NEB and dimer methods, in the same way as it was done in Chapter 4 and in [16]. We saw in our results that the momentum descent method for saddle point search methods compares in general to the *ode12r* adaptive step selection but it has a much simpler and more intuitive formulation to *ode12r*. It is also evident that especially in the unpreconditioned setting, but also to some extent in the preconditioned setting, the method could benefit from an adaptive step length selection similar to the *ode12r* scheme. This is a more difficult problem as one needs to adaptively choose the momentum parameter and step length simultaneously, since even though some intuition has been given on how to select a good momentum coefficient, it should not be selected independently of the step length. In the preconditioned case, we have seen that the momentum coefficient is in fact≈ 2 simplifying matters, however this depends on the accuracy of the preconditioner and one should not expect that this is always the case. This goes back to our previous comment on the necessity of constructing improved preconditioners.

In conclusion, this thesis has incorporated simple ideas from numerical analysis (for the most part numerical optimisation) in saddle point finding methods, to accelerate their convergence. The methodology is general so that any scheme in the family of MEP finding methods and dimer methods may be used straight-forwardly (see for example *simplified string* [11] and *improved tangent NEB* [14]). Adaptivity is actually possible also in the case of hybrid models (see for example hybrid QM/MM models [76]). The results are promising, offering good convergence speedups and improving the robustness of the schemes, but also suggesting a range of new directions where work may be conducted in to improve these results even further.

All schemes developed in this thesis have been coded in the `Julia` programming language and are available in `https://github.com/cortner/SaddleSearch.jl`. In our next steps we wish to port this code to ASE for wider use.

Appendix A

Algebraic Calculations for Chapter 5

In Sec. 5.2, we defined the curves $g^{\pm}(b)$ defined on the interval $b \in [0, \infty)$ and given by equation (5.14). In this appendix we will look at the asymptotic behaviour of g^{+} as $b \to 0^{+}$ and as $b \to \infty$ and find the fixed point of g^{+}. The expression for g^{+} may be re-written as

$$g^{+}(b) = b - \frac{\sqrt{2}}{2}\Big(\big[(b^2 - \alpha)^2 + \gamma^2\big]^{\frac{1}{2}} + b^2 - \alpha \Big)^{\frac{1}{2}} \tag{A.1}$$

where we have assigned the constants $\alpha := 4\mathrm{Re}(\lambda) > 0$ and $\gamma := 4\mathrm{Im}(\lambda)$ (β is reserved for later on).

A.1 Asymptotic behaviour at zero

First we look at the the asymptotic behaviour of g^+ as $b \to 0^+$:

$$\begin{aligned}
g^+(b) &= b - \frac{\sqrt{2}}{2}((b^4 - 2\alpha b^2 + \alpha^2 + \gamma^2)^{\frac{1}{2}} + b^2 - \alpha)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(4|\lambda|\left(1 - \frac{\alpha}{8|\lambda|^2}b^2 + \frac{1}{16|\lambda|^2}b^4\right)^{\frac{1}{2}} + b^2 - \alpha\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(4|\lambda| - \alpha + \left(1 - \frac{\alpha}{4|\lambda|}\right)b^2 + \mathcal{O}(b^4)\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}(4|\lambda| - \alpha)^{\frac{1}{2}}\left(1 + \frac{1}{4|\lambda|}b^2 + \mathcal{O}(b^4)\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}(4|\lambda| - \alpha)^{\frac{1}{2}}\left(1 + \frac{1}{8|\lambda|}b^2 + \mathcal{O}(b^4)\right) \\
&= -\frac{\sqrt{2}}{2}(4|\lambda| - \alpha)^{\frac{1}{2}} + b + \mathcal{O}(b^2)
\end{aligned} \tag{A.2}$$

A.2 Asymptotics while approaching infinity

Looking at the asymptotic behaviour of $b \to \infty$, g^+ behaves like

$$\begin{aligned}
g^+(b) &= b - \frac{\sqrt{2}}{2}((b^4 - 2\alpha b^2 + \alpha^2 + \gamma^2)^{\frac{1}{2}} + b^2 - \alpha)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(b^2(1 - 2\alpha b^{-2})^{\frac{1}{2}}\left(1 + b^{-4}(1 - 2\alpha b^{-2})^{-1}|\lambda|^2\right)^{\frac{1}{2}} + b^2 - \alpha\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(b^2\Big(1 - \alpha b^{-2} + \mathcal{O}(b^{-4})\Big)\Big(1 + \mathcal{O}(b^{-4})\Big) + b^2 - \alpha\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(b^2\Big(1 - \alpha b^{-2} + \mathcal{O}(b^{-4})\Big) + b^2 - \alpha\right)^{\frac{1}{2}} \\
&= b - \frac{\sqrt{2}}{2}\left(2b^2 - 2\alpha + \mathcal{O}(b^{-2})\right)^{\frac{1}{2}} \\
&= b - b\big(1 - \alpha b^{-2} + \mathcal{O}(b^{-4})\big)^{\frac{1}{2}} \\
&= \frac{1}{2}\alpha b^{-1} + \mathcal{O}(b^{-3}).
\end{aligned} \tag{A.3}$$

A.3 Classification of fixed points

Finally, we are interested in the fixed points of g^+ in $[0, \infty)$ The first derivative of g^+ is

$$\frac{\partial g^+}{\partial b} = 1 - \frac{\sqrt{2}}{2}\frac{\partial}{\partial b}\Big(\big[(b^2-\alpha)^2+\gamma^2\big]^{\frac{1}{2}} + b^2 - \alpha\Big)^{\frac{1}{2}} \tag{A.4}$$

Let $\beta = b^2 - \alpha$,

$$\frac{\partial g^+}{\partial b} = 1 - \frac{\sqrt{2}}{2}\frac{\partial}{\partial \beta}\left[\Big(\big[\beta^2+\gamma^2\big]^{\frac{1}{2}} + \beta\Big)^{\frac{1}{2}}\right]\frac{\partial \beta}{\partial b} \tag{A.5}$$

$$= 1 - \frac{\sqrt{2}}{2}\sqrt{\beta+\alpha}\Big(\big[\beta^2+\gamma^2\big]^{\frac{1}{2}} + \beta\Big)^{-\frac{1}{2}}\Big(\beta\big[\beta^2+\gamma^2\big]^{-\frac{1}{2}} + 1\Big). \tag{A.6}$$

Equating $\partial g^+/\partial b = 0$ gives

$$\begin{aligned}
&2 = \sqrt{2}\sqrt{\beta+\alpha}\Big(\big[\beta^2+\gamma^2\big]^{\frac{1}{2}} + \beta\Big)^{-\frac{1}{2}}\Big(\beta\big[\beta^2+\gamma^2\big]^{-\frac{1}{2}} + 1\Big)\\
&\Rightarrow 2\big(\big[\beta^2+\gamma^2\big]^{\frac{1}{2}} + \beta\big) = (\beta+\alpha)\big[\beta^2+\gamma^2\big]^{-1}\big(\beta + \big[\beta^2+\gamma^2\big]^{\frac{1}{2}}\big)^2\\
&\Rightarrow 2\big(\beta^2+\gamma^2\big) = (\beta+\alpha)\big(\beta + \big[\beta^2+\gamma^2\big]^{\frac{1}{2}}\big)\\
&\Rightarrow 2\beta^2 + 2\gamma^2 = \beta^2 + \alpha\beta + (\beta+\alpha)\sqrt{\beta^2+\gamma^2}\\
&\Rightarrow \big(\beta^2 - \alpha\beta + 2\gamma^2\big)^2 = (\beta^2 + 2\alpha\beta + \alpha^2)(\beta^2+\gamma^2)\\
&\Rightarrow \beta^4 - 2\alpha\beta^3 + (\alpha^2+4\gamma^2)\beta^2 - 4\alpha\gamma^2\beta + 4\gamma^4\\
&\quad = \beta^4 + 2\alpha\beta^3 + (\alpha^2+\gamma^2)\beta^2 + 2\alpha\gamma^2\beta + \alpha^2\gamma^2\\
&\Rightarrow 4\alpha\beta^3 - 3\gamma^2\beta^2 + 6\alpha\gamma^2\beta + (\alpha^2\gamma^2 - 4\gamma^4) = 0.
\end{aligned} \tag{A.7}$$

Therefore, the fixed points b of g^+ are given by

$$b = \sqrt{\beta+\alpha} \tag{A.8}$$

where β are roots of the cubic polynomial

$$q(\beta) = 4\alpha\beta^3 - 3\gamma^2\beta^2 + 6\alpha\gamma^2\beta + (\alpha^2\gamma^2 - 4\gamma^4). \tag{A.9}$$

The discriminant Δ of q is given by

$$\begin{aligned}\Delta =&(-3\gamma^2)^2(6\alpha\gamma^2) - 4(4\alpha)(6\alpha\gamma^2)^3 - 4(-3\gamma^2)^3(\alpha^2\gamma^2 - 4\gamma^4)\\ &- 27(4\alpha)^2(\alpha^2\gamma^2 - 4\gamma^4) + 18(4\alpha)(-3\gamma^2)(6\alpha\gamma^2)(\alpha^2\gamma^2 - 4\gamma^4)\\ &= 324\alpha^2\gamma^8 - 3456\alpha^4\gamma^6 + 108\alpha^2\gamma^8 - 432\gamma^{10}\\ &- 432\alpha^6\gamma^4 + 3456\alpha^4\gamma^6 - 6912\alpha^2\gamma^8\\ &+ 1296\alpha^4\gamma^6 + 5184\alpha^2\gamma^8\\ &= -432\gamma^{10} - 432\alpha^6\gamma^4 - 1296\alpha^2\gamma^8 - 1296\alpha^4\gamma^6 < 0, \quad \forall\gamma \neq 0 \end{aligned} \tag{A.10}$$

Thus, q has a single real root. This may be derived algebraically for example from Cardano's formula [82]. From equations (A.8), (A.9) and (A.10), we conclude that (A.1) has a single fixed point.

Bibliography

[1] M.-C. Marinica, L. Ventelon, M. R. Gilbert, L. Proville, S. L. S L Dudarev, J. Marian, G. Bencteux, and F. Willaime. Interatomic potentials for modelling radiation defects and dislocations in tungsten. *Journal of Physics: Condensed Matter*, 25(39):395502, 2013.

[2] L. Mones, C. Ortner, and G. Csanyi. Preconditioners for the geometry optimisation and saddle point search of molecular systems. *Scientific Reports*, 8(1):13991, 2018.

[3] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E Castelli, R. Christensen, M. Dułak, J. Friis, M. N Groves, B. Hammer, C. Hargus, E. D Hermes, P. C Jennings, P. B. Jensen, J. Kermode, J. R Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W Jacobsen. The atomic simulation environment—a python library for working with atoms. *J. Phys. Condens. Matter*, 29(27):273002, June 2017.

[4] E. Pollak and P. Talkner. Reaction rate theory: What it was, where is it today, and where is it going? *Chaos*, 15(2):26116, 2005.

[5] A. F. Voter. Introduction to the kinetic monte carlo method. *Radiation Effects in Solids. NATO Science Series*, 235:1–23, 2007.

[6] S. Arrhenius. Über die dissociationswärme und den einfluss der temperatur auf den dissociationsgrad der elektrolyte. *Z. Phys. Chem.*, 4:96–116, 1889.

[7] H. Eyring. The Activated Complex in Chemical Reactions. *Journal of Computational Physics*, 3:107–115, 1935.

[8] G. H. Vineyard. Frequency factors and isotope effects in solid state rate processes. *Journal of Physics and Chemistry of Solids*, 3(1):121–127, 1957.

[9] G. Henkelman and H. Jónsson. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *The Journal of Chemical Physics*, 111(15):7010–7022, 1999.

[10] W. E, W. Ren, and E. Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66:052301, Aug 2002.

[11] W. E, W. Ren, and E. Vanden-Eijnden. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *The Journal of Chemical Physics*, 126(16), 2007.

[12] M. Cameron, R. V. Kohn, and E. Vanden-Eijnden. The string method as a dynamical system. *Journal of Nonlinear Science*, 21(2):193–230, 2011.

[13] H. Jónsson, G. Mills, and K. W. Jacobsen. Nudged elastic band method for finding minimum energy paths of transitions. *Classical and Quantum Dynamics in Condensed Phase Simulations*, pages 385–404, June 1998.

[14] G. Henkelman and H. Jónsson. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *The Journal of Chemical Physics*, 113(22):9978–9985, 2000.

[15] J. Nocedal and S. J. Wright. Springer Series in Operations Research and Financial Engineering. Springer, Berlin, 2006.

[16] N. Gould, C. Ortner, and D. Packwood. A dimer-type saddle search algorithm with preconditioning and linesearch. *Math. Comp.*, 85:2939–2966, 2016.

[17] N. W. Ashcroft and N. D. Mermin. *Solid State Physics*. HRW international editions. Holt, Rinehart and Winston, 1976.

[18] M. I. Aroyo, U. Müller, and H. Wondratschek. *Historical introduction*, pages 2–5. Springer Netherlands, Dordrecht, 2004.

[19] K. Müller and L. D. Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica chimica acta*, 53(1):75–93, Mar 1979.

[20] E. Fermi, J. Pasta, and S. Ulam. Los alamos national laboratory report (la1940), also in collected papers of enrico fermi 2, 1955.

[21] J. E. Lennard-Jones and S. Chapman. On the determination of molecular fields. i. from the variation of the viscosity of a gas with temperature. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106(738):441–462, 1924.

[22] J. E Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43(5):461–482, 1931.

[23] P. M. Morse. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Phys. Rev.*, 34:57–64, Jul 1929.

[24] M. S. Daw, S. M. Foiles, and M. I. Baskes. The embedded-atom method: a review of theory and applications. *Materials Science Reports*, 9(7):251 – 310, 1993.

[25] J. B. Adams. Bonding energy models. In K. H. Jürgen Buschow, Robert W. Cahn, Merton C. Flemings, Bernhard Ilschner, Edward J. Kramer, Subhash Mahajan, and Patrick Veyssière, editors, *Encyclopedia of Materials: Science and Technology*, pages 763 – 767. Elsevier, Oxford, 2001.

[26] M. S. Daw and M. I. Baskes. Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals. *Phys. Rev. Lett.*, 50:1285–1288, Apr 1983.

[27] M. S. Daw and M. I. Baskes. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B*, 29:6443–6453, Jun 1984.

[28] M. W. Finnis and J. E. Sinclair. A simple empirical n-body potential for transition metals. *Philosophical Magazine A*, 50(1):45–55, 1984.

[29] R. P. Gupta. Lattice relaxation at a metal surface. *Phys. Rev. B*, 23:6265–6270, Jun 1981.

[30] W. Schommers. Chapter 2 - Theoretical and computational methods. In W. Schommers, editor, *Basic Physics of Nanoscience*, pages 91 – 202. Elsevier, 2019.

[31] F. Ercolessi and J. B. Adams. Interatomic potentials from first-principles calculations: The force-matching method. *Europhysics Letters (EPL)*, 26(8):583–588, 1994.

[32] M. A. Tschopp, D. E. Spearot, and D. L. McDowell. Chapter 82 - influence of grain boundary structure on dislocation nucleation in fcc metals. In J. P. Hirth, editor, *A Tribute to F.R.N. Nabarro*, volume 14 of *Dislocations in Solids*, pages 43 – 139. Elsevier, 2008.

[33] G. J. Ackland and R. Thetford. An improved n-body semi-empirical model for body-centred cubic transition metals. *Philosophical Magazine A*, 56(1):15–30, 1987.

[34] M. I. Mendelev, S. Han, D. J. Srolovitz, G. J. Ackland, D. Y. Sun, and M. Asta. Development of new interatomic potentials appropriate for crystalline and liquid iron. *Philosophical Magazine*, 83(35):3977–3994, 2003.

[35] G. J. Ackland, M. I. Mendelev, D. J. Srolovitz, S. Han, and A. V. Barashev. Development of an interatomic potential for phosphorus impurities in α-iron. *Journal of Physics: Condensed Matter*, 16(27):S2629–S2642, jun 2004.

[36] P. A. Gordon, T. Neeraj, and M. I. Mendelev. Screw dislocation mobility in bcc metals: a refined potential description for α-fe. *Philosophical Magazine*, 91(30):3931–3945, 2011.

[37] L. Proville, D. Rodney, and M.-C. Marinica. Quantum effect on thermally activated glide of dislocations. *Nature Materials*, 11:845–849, 08 2012.

[38] M. Born and R. Oppenheimer. Zur Quantentheorie der Molekeln. *Annalen der Physik*, 389:457–484, 1927.

[39] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[40] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[41] K. Burke, J. P. Perdew, and Y. Wang. *Derivation of a Generalized Gradient Approximation: The PW91 Density Functional*, pages 81–111. Springer US, Boston, MA, 1998.

[42] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[43] J. E. Huheey, E. A. Keiter, R. L. Keiter, and O. K. Medhi. *Inorganic Chemistry: Principles of Structure and Reactivity.* Pearson Education, 2006.

[44] S. Clark, M. Segall, C. Pickard, P. Hasnip, M. Probert, K. Refson, and M. Payne. First principles methods using castep. *Zeitschrift für Kristallographie - Crystalline Materials*, 220:567–570, 2005.

[45] F. Bloch. Über die quantenmechanik der elektronen in kristallgittern. *Zeitschrift für Physik*, 52(7):555–600, Jul 1929.

[46] H. J. Monkhorst and J. D. Pack. Special points for brillouin-zone integrations. *Phys. Rev. B*, 13:5188–5192, Jun 1976.

[47] C. Ortner. Julip: Julia library for interatomic potentials.

[48] Private communication with P. Grigorev.

[49] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1 – 19, 1995.

[50] D. Frenkel and B. Smit. *Understanding Molecular Simulation (Second Edition).* Academic Press, San Diego, second edition edition, 2002.

[51] A. F. Voter, F. Montalenti, and T. C. Germann. Extending the time scale in atomistic simulation of materials. *Annual Review of Materials Research*, 32(1):321–346, 2002.

[52] W. E and X. Zhou. The gentlest ascent dynamics. *Nonlinearity*, 24(6):1831, 2011.

[53] G. Díaz Leines and J. Rogal. Comparison of minimum-action and steepest-descent paths in gradient systems. *Phys. Rev. E*, 93:022307, Feb 2016.

[54] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989.

[55] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–436, 1952.

[56] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch. Structural Relaxation Made Simple. *Physical Review Letters*, 97(17):170201, 2006.

[57] B. T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[58] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[59] P. Dierckx. Oxford University Press, Inc., New York, NY, USA, 1993.

[60] G. Henkelman, B. P. Uberuaga, and H. Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics*, 113(22):9901–9904, 2000.

[61] J. Zhang and Q. Du. Shrinking dimer dynamics and its applications to saddle point search. *SIAM Journal on Numerical Analysis*, 50(4):1899–1921, 2012.

[62] R. A. Olsen, G. J. Kroes, G. Henkelman, A. Arnaldsson, and H. Jónsson. Comparison of methods for finding saddle points without knowledge of

the final states. *The Journal of Chemical Physics*, 121(20):9776–9792, 2004.

[63] J. A. Garrido Torres, P. C. Jennings, M. H. Hansen, J. R. Boes, and T. Bligaard. Low-scaling algorithm for nudged elastic band calculations using a surrogate machine learning model. *Phys. Rev. Lett.*, 122:156001, Apr 2019.

[64] A. Samanta and W. E. Optimization-based string method for finding minimum energy path. *Communications in Computational Physics*, 14(2):265–275, 008 2013.

[65] E. Hairer, S. P. Nørsett, and G. Wanner. Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[66] A. M. Stuart. Probabilistic and deterministic convergence proofs for software for initial value problems. *Numerical Algorithms*, 14(1):227–260, 1997.

[67] H. Lamba and A. M. Stuart. Convergence results for the matlab ode23 routine. *BIT Numerical Mathematics*, 38(4):751–780, 1998.

[68] H. Lamba. Dynamical systems and adaptive timestepping in ode solvers. *BIT Numerical Mathematics*, 40(2):314–335, 2000.

[69] P. K. Mogensen and A. N. Riseth. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615, 2018.

[70] S. Makri, C. Ortner, and J. R. Kermode. A preconditioning scheme for minimum energy path finding methods. *The Journal of Chemical Physics*, 150(9):094109, 2019.

[71] D. Packwood, J. Kermode, L. Mones, N. Bernstein, J. Woolley, N. Gould, C. Ortner, and G. Csányi. A universal preconditioner for simulating condensed phase materials. *The Journal of Chemical Physics*, 144(16):164109, April 2016.

[72] R. Lindh, A. Bernhardsson, G. Karlström, and P.-Å Malmqvist. On the use of a hessian model function in molecular geometry optimizations. *Chemical Physics Letters*, 241(4):423 – 428, 1995.

[73] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. *Reviews of Modern Physics*, 64:1045–1097, 1992.

[74] H. Chen and C. Ortner. QM/MM methods for crystalline defects. part 2: Consistent energy and force-mixing. *Multiscale Modeling & Simulation*, 15(1):184–214, 2017.

[75] H. Chen and C. Ortner. QM/MM methods for crystalline defects. part 1: Locality of the tight binding model. *Multiscale Modeling & Simulation*, 14(1):232–264, 2016.

[76] N. Bernstein, J. R. Kermode, and G. Csányi. Hybrid atomistic simulation methods for materials systems. *Reports on Progress in Physics*, 72(2):026501, 2009.

[77] S. Weijie, B. Stephen, and J. C. Emmanuel. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

[78] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[79] J. Calder and A. Yezzi. PDE Acceleration: A convergence rate analysis and applications to obstacle problems. *Research in the Mathematical Sciences*, 6(4):35, 2019.

[80] M. Benyamin, J. Calder, G. Sundaramoorthi, and A. Yezzi. Accelerated PDE's for efficient solution of regularized inversion problems. *Journal of Mathematical Imaging and Vision*, 62(1):10–36, 2020.

[81] M. I. J. Probert. Improved algorithm for geometry optimisation using damped molecular dynamics. *Journal of Computational Physics*, 191(1):130 – 146, 2003.

[82] G. Cardano and T. R. Witmer. *Ars Magna Or The Rules of Algebra*. Dover Books on Advanced Mathematics. Dover, 1968.